

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

CAD SYSTÉM PRO 2D KRESLENÍ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MICHAL MURÍN

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

CAD SYSTÉM PRO 2D KRESLENÍ

CAD SYSTEM FOR 2D DRAWING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAL MURÍN

VEDOUcí PRÁCE

SUPERVISOR

doc. Ing. PŘEMYSL KRŠEK, Ph.D.

BRNO 2011

Abstrakt

Bakalářská práce je zaměřená na návrh a tvorbu CAD systému pro dvojrozměrné kreslení. Obeznamuje s využitím počítačů v průmyselné výrobě a s výhodami, které jejich využití přináší, s obecnou definicí CAD systémů a požadavkami na ně. Práce obsahuje návrh vytvářeného CAD systému a jako praktickou část, implementaci navržené aplikace. Na závěr je shrnuto, čeho bylo při tvorbě dosaženo a čeho by potenciálně v budoucnu mohlo být dosaženo při dalším rozšiřování aplikace.

Abstract

The Bachelor's thesis is aimed at the design and creation of CAD system for 2D drawing. It introduces the use of computers in manufacturing and the advantages the use of computers brings, general definition of CAD systems and requirements imposed on them. The thesis contains the design of generated CAD system, and as a practical part it introduces implementation of the designed application. Finally it summarizes what has been achieved in developing and what could potentially be achieved in further expansion of the application in the future.

Klíčová slova

CAD, CAM, vektorová grafika, 2D kreslení, OpenGL

Keywords

CAD, CAM, vector graphics, 2D drawing, OpenGL

Citace

Michal Murín: CAD systém pro 2D kreslení, bakalářská práce, Brno, FIT VUT v Brně, 2011

CAD systém pro 2D kreslení

Prohlášení

Prohlašuji, že jsem tuhle bakalářskou práci vypracoval samostatně pod vedením pána doc. Ing. Přemysla Krška, PhD. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Michal Murín
17. mája 2011

Poděkování

Rád bych se poděkoval vedoucímu této práce doc. Ing. Přemyslovi Krškovi, PhD., který mi počas konzultací poskytoval své odborné rady a připomínky, čím přispěl k vypracování této práce.

© Michal Murín, 2011.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
2 Teoretický rozbor problematiky	4
2.1 Počítače v priemyselnej výrobe	4
2.2 CAD systém	5
2.2.1 Návrh modelu	6
2.2.2 Rozdelenie dnešných CAD systémov podľa ich rozsahu	7
2.3 Integrované systémy	7
2.3.1 CAM	7
2.3.2 CAE	9
2.3.3 CAPP	9
2.3.4 CIM	9
2.4 Požiadavky na CAD systém	10
2.4.1 Presnosť	10
2.4.2 Organizácia	12
2.4.3 Reprezentácia dát v 2D CAD systéme	12
2.4.4 Prenositelnosť výkresov medzi rôznymi CAD systémami	12
2.4.5 Uživatelské rozhranie v CAD systémoch	13
3 Návrh a implementácia aplikácie	14
3.1 Ciele	14
3.2 Dátová reprezentácia	15
3.2.1 Entity	15
3.2.2 Dátové kontajnery – hladiny a plátna	18
3.3 Spôsob zobrazovania	19
3.4 Významné funkcie aplikácie	20
3.4.1 Transformácie plátna	20
3.4.2 Výberový mód	21
3.4.3 Uchopovací mód	23
3.4.4 Transformácie vybraných objektov	24
3.4.5 Ortogonálny mód	26
3.4.6 Práca so súborom	26
3.5 Grafické uživatelské rozhranie	27
3.5.1 Scéna, výber entity a štýly čiar	27
3.5.2 Zadávanie parametrov klávesnicou	28
3.5.3 Transformácie výberu	28
3.5.4 Tabuľka hladín	29
3.5.5 Uchopovací mód	30

3.5.6	Ortogonalný mód	30
3.5.7	Priblíženie plátna	30
3.6	Implementačné prostriedky	31
3.6.1	OpenGL	31
3.6.2	Qt	31
4	Výsledky	32
4.1	Systém ako aplikácia	32
4.2	Entity a dátové kontajnery	33
4.3	Transformácie plátna	34
4.4	Výberový mód	34
4.5	Transformácie výberu	34
4.6	Uchopovací mód	34
4.7	Ukladanie a načítanie práce	35
5	Záver	36
5.1	Možnosti budúcej práce	36
A	Obsah CD	39

Kapitola 1

Úvod

Moderná doba ponúka ľuďom široké možnosti v oblasti počítačových technológií, ktoré prenikajú do mnohých odvetví priemyslu. Takýmito odvetviami sú napríklad architektúra a strojný priemysel, ktoré by sa nemohli zaobiť bez technických či architektonických návrhov, nákresov a projektov.

Kedysi ručné kreslenie návrhov a nákresov bolo nahradené kreslením počítačovým, najmä pomocou systémov CAD (Computer Aided Design). Hlavnými výhodami CAD systémov pri vytváraní nákresov sú jednoduchosť práce, presnosť nákresov a v neposlednom rade rýchlosť ich tvorby. Dnešné CAD systémy disponujú širokou škálou funkcií a sú neustále rozširované a zdokonaľované. Ponúkajú možnosti tvorby ako dvojrozmerných, napr. pôdorys budovy, tak aj trojrozmerných, napr. model súčiastky, návrhov. Existuje rada špecializovaných systémov, ktoré sú zamerané na tvorbu špecifických návrhov, ako napr. elektrických obvodov, diagramov tried (softwarové inžinierstvo), a pod. Medzi najznámejšie CAD systémy patrí napr. AutoCAD.

Táto práca je zameraná konkrétne na CAD systém pre kreslenie dvojrozmerných nákresov. V nasledujúcej kapitole oboznamuje čitateľa s využitím počítačov v priemyselnej výrobe a s výhodami, ktoré toto spojenie prináša a definíciou CAD systému. V tejto kapitole sú následne rozobrané základné požiadavky na funkčnosť a užívateľské rozhranie CAD systému pre kreslenie dvojrozmerných nákresov. Ďalšie kapitoly obsahujú návrh a implementáciu aplikácie, popis použitých postupov a metód a alternatívnych možností tvorby niektorých častí aplikácie a dosiahnuté ciele. V závere je uvedené komplexné zhodnotenie výsledkov práce, jej prínosu a možnosti budúcich rozšírení aplikácie.

Kapitola 2

Teoretický rozbor problematiky

Kapitola predstavuje teoretickú časť práce, ktorá rozoberá použitie počítačov vo výrobe, vysvetľuje význam pojmov spojených s danou problematikou. Informácie podané v tejto kapitole vychádzajú z knihy [11], učebných materiálov k predmetu Vizualizace a CAD [4], [5], [6], [7], opory k predmetu Základy počítačovej grafiky [3] a skript Úvod do počítačovej grafiky a CAD systémov [1]. Nakoniec charakterizuje vlastnosti dnešných CAD systémov a požiadavky na ne, na základe ktorých budú v nasledujúcej kapitole stanovené ciele a uvádza príklady dnes používaných CAD systémov.

2.1 Počítače v priemyselnej výrobe

Úloha počítačov vo výrobe môže byť rozdelená do 2 kategórií:

1. Počítačové monitorovanie a kontrola výrobného procesu
2. Výrobu podporujúce aplikácie, ktoré sa v podstate zaoberajú prípravami na samotnú výrobu a povýrobnými operáciami.

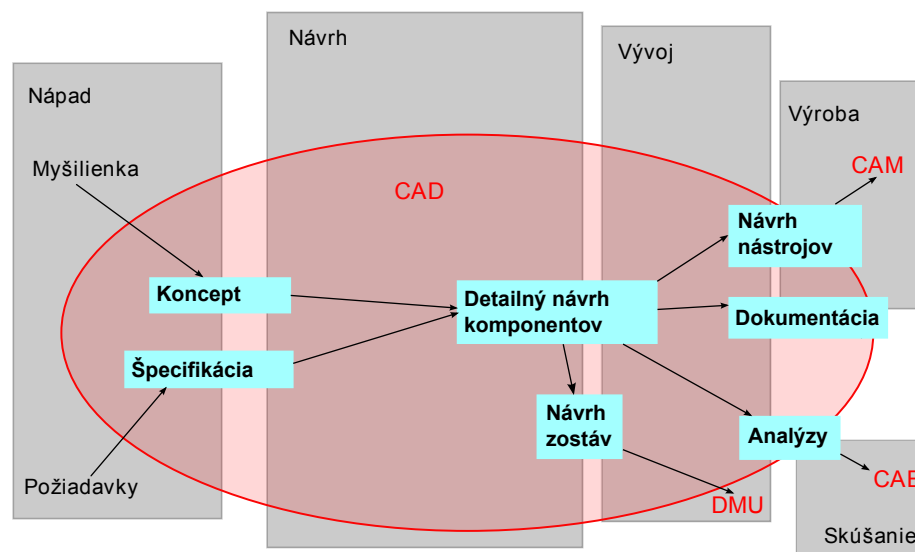
Pre túto prácu je zaujímavá 2. kategória zahŕňajúca podporné funkcie, ktoré môžu počítače poskytovať pre úspešné vykonávanie výrobných operácií. Uvažované typy podpory:

- CAD – computer aided design. Používa také počítačové metódy na vytvorenie geometrického modelu produktu v 3D forme, že je možné posudzovať geometrické a výrobné požiadavky.
- CADD – computer aided design and drafting. Kombinácia CAD funkcií a návrhu pri generovaní výrobných výkresov za účelom následného spracovania.
- CAE – computer aided engineering. Použitie počítačových metód na podporu základnej kontroly chýb, analýzy, optimalizácie, atď., návrhu produktu.
- CAM – computer aided manufacturing. Obecne sa jedná o počítačový software používaný na tvorbu CNC¹ častí programov pre výrobu a iné aplikácie spracovania.
- CAPP – computer aided process planning. Použité počítača pri tvorbe plánov spracovania pre kompletnú výrobu produktu a jeho častí.

¹Computer Numerical Control (počítačové číslicové riadenie)

- CATD – computer aided tool design. Počítačová podpora využitá pri vývoji nástrojov výroby ako, napr. príslušenstvo a šablóny.
- CAP – computer aided planning. Použitie počítača pre plánovanie.
- CAQ – computer aided quality. Počítačová podpora riadenia kvality produkcie.

Použitie počítačov vo výrobe je metodický prístup k podnikaniu s účelom zlepšenia priemyselnej výkonnosti. To vyžaduje rad technológií, ktoré sa stali realizovateľnými vďaka rozvoju počítačov. Celý proces vývoja produktov prostredníctvom CAD systémov je schematicky znázornený na obrázku 2.4.



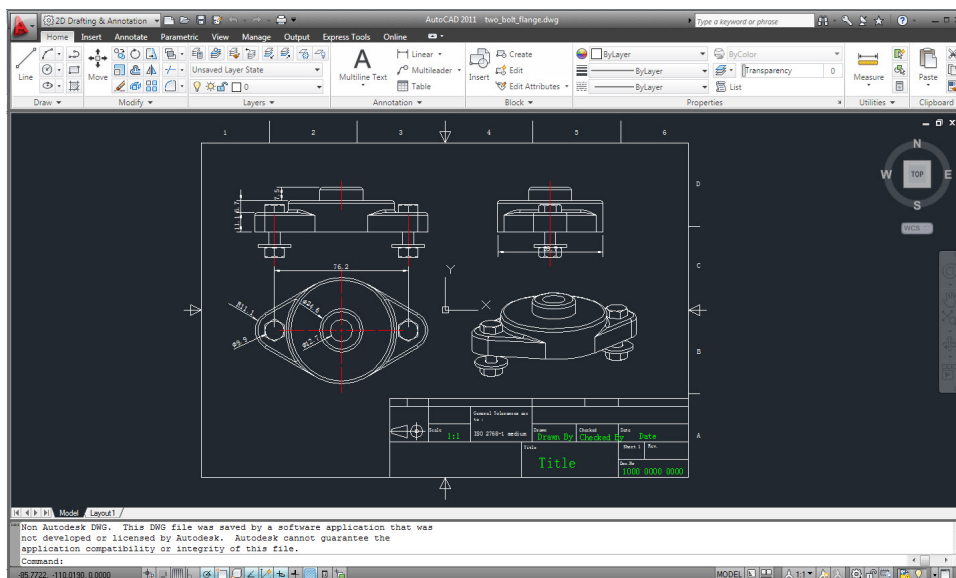
Obrázok 2.1: Vývoj produktu prostredníctvom CAD systémov [8]

2.2 CAD systém

Skratka CAD vychádza z anglického názvu Computer Aided Design, ktorý môže byť preložený ako počítačom podporované konštruovanie. CAD systémy sú úzko späté s výrobou, vytvárajú presný popis toho, čo a ako sa má vyrábať (stavať). Hlavným komunikačným prostriedkom je dnes stále výkresová dokumentácia, ktorej 2D podoba je výsledkom práce s CAD systémom. Dnes sú využívané v širokom spektre odvetví – stavebníctvo a architektúra, zdravotníctvo, strojný priemysel, filmový priemysel a mnohé iné, pričom ich využitie a ponuka sa neustále rozrastajú. CAD je určený pre ľudí s tvorivou profesiou – konštruktéri, návrhári, projektanti, architekti, vývojári, dizajnéri a iní.

CAD systém je vektorový grafický kresliaci a modelovací konštrukčný nástroj pre tvorbu grafickej výrobnej dokumentácie. Dá sa naň pozerať 2 pohľadmi:

- použitia – grafický, kresliaci program, konštrukčný, presný nástroj, 2D/3D systém
- počítačovej grafiky – grafický, vektorový program/editor, databáza vektorových entít



Obrázok 2.2: AutoCAD 2011 od firmy Autodesk

2.2.1 Návrh modelu

Návrh modelu je aktivita, ktorá potrebuje byť dobre organizovaná a musí dodržiavať podmienky, ktoré zaručia úspešný vývoj produktu. Pod pojmom produkt rozumieme komponentu, ktorá je funkčnou množinou veľkého množstva menších častí, z ktorých každá je nevyhnutná pre fungovanie celku. Komplexnosť návrhu vzrastá s počtom komponent reprezentujúcich konečnú časť.

Dnešné CAD technológie poskytujú pri vytváraní návrhu nasledujúce výhody:

- CAD je rýchlejší a presnejší ako konvenčné metódy.
- Rôzne možnosti konštruovania dostupné v CAD systéme výrazne uľahčujú navrhovanie a vytváranie modelu.
- V porovnaní s klasickými metódami kreslenia, CAD poskytuje možnosť manipulácie s dimenziami, vlastnosťami a rozmermi kreslených objektov. To robí CAD veľmi vhodným nástrojom pre návrh modelov.
- S CAD systémom nie je nutné opakovať návrh alebo nakreslenie ľubovoľnej komponenty, keď už je raz nakreslená. Môže byť kopírovaná do iných prác, zahŕňajúc tiež potrebné geometrické transformácie.
- Môžeme presne počítať rôzne geometrické vlastnosti vrátane dimenzií komponent.
- Modifikácia modelu je veľmi jednoduchá, robí úlohu návrhára jednoduchšou a dáva mu možnosť vytvoriť návrh, ktorý bude v budúcnosti jednoducho rozšíriteľný.
- Možnosť používať štandardné komponenty (tzv. „part libraries“) zrýchľuje návrh modelu. Veľké množstvo menších komponent a podmnožín návrhu môže byť uložených v „part libraries“ pre ich neskoršie znovupoužitie v inej práci.

- Niekoľko profesionálnych CAD balíkov zahŕňa trojrozmernú vizualizáciu, takže návrhári môžu nahliadať na vytvorený model z rôznych pohľadov.

2.2.2 Rozdelenie dnešných CAD systémov podľa ich rozsahu

Malé CAD systémy

Relatívne lacné, riešiacie úplné skicovanie a kreslenie (tvorba náčrtov), ale nie sú určené pre konštruktérov. Ide o programy, ktoré nezachádzajú za hranicu 2D.

Stredné CAD systémy

Podporujú úplný 2D a čiastočný 3D design. Pracujú v rovine a majú na vysokej úrovni prepracované 2D kresliace nástroje (bod, úsečka, oblúk, kružnica, prsteň, elipsa, mnoho-uholník, krivka, dvojité čiary a pod.), uchyťavacie režimy (stred, koncový bod, tangenta, kolmica, priesečník a kvadrant), editovacie nástroje (pole, kópia, zrkadlenie, zaoblenie, skosenie, posun, orezávanie, predĺženie, mierka, natiahnutie, otočenie), možnosť práce v hľadinách, veľké množstvo typov čiar a farieb. Sú doplnené o niektoré 3D nástroje, ako napr. vytvorenie 3D modelu z tvoriacej krivky rotáciou či posunutím, prácu s B-spline krivkami a plochami, premietnutie krivky na určenú rovinu, konštrukciu špirálovej plochy a iné. S možnosťami týchto systémov rastú aj ich nároky na hardware a zvyšuje sa samozrejme aj ich cena.

Veľké CAD systémy

Software pracujúci v priestore, t.j. v 3D. Majú analogické kresliace a uchyťavacie nástroje, editovacie režimy ako predchádzajúce, s tým rozdielom, že sú to nástroje pracujúce plne v 3D priestore. Príkladom môže byť PowerShape, SolidWorks a iné. Disponujú najprepracovanejšími a najvýkonnejšími modelovacími technikami alebo ako sa zvykne zjednodušene hovoriť, výkonnými modelármi, ktoré dovoľujú skutočné počítačové modelovanie. Mnohé výkonné CAD software sú tiež súčasťou veľkých CAD/CAM software.

2.3 Integrované systémy

2.3.1 CAM

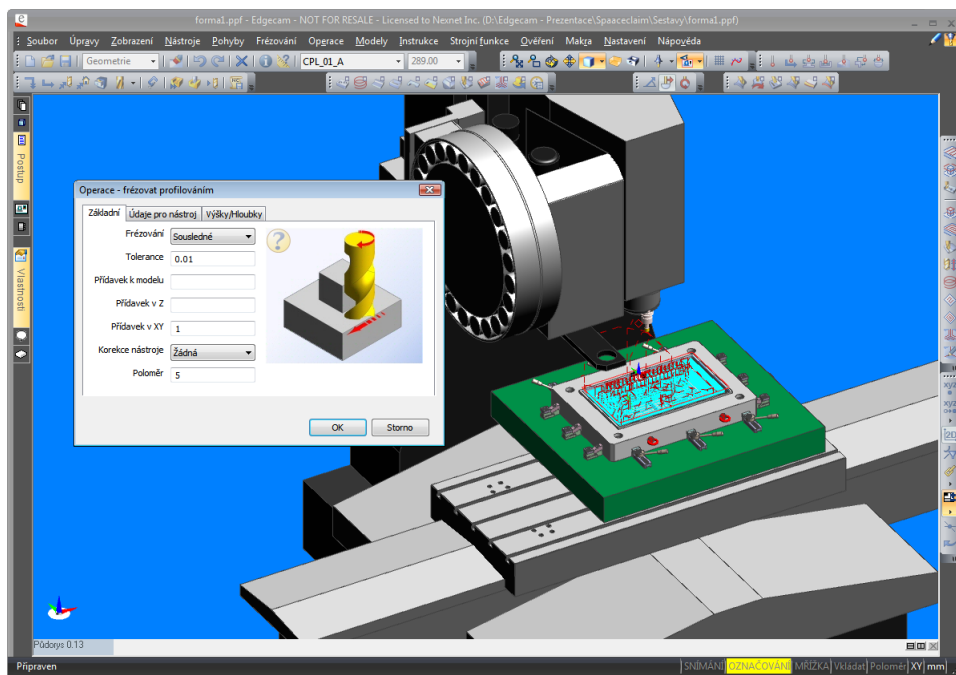
Skratku CAM – Computer Aided Manufacturing môžeme preložiť ako počítačom podporovaná výroba. CAM systémy zahŕňajú počítačové číslicové riadenie (CNC) výrobnej techniky, robotov, medzioperačnej dopravy výrobkov, polotovarov, náradia a pod. Patrí sem počítačové operatívne riadenie výroby na dielenskej úrovni, číslicovo riadené výrobné systémy, automatizované dopravníky, automatizované sklady a pod. CAM je možné chápať v dvoch úrovniach – ako nejaký konkrétny systém (CNC riadenie a NC stroj), alebo ako určitý komplexný pohľad na počítačovú podporu vo výrobe. Pod CAM si môžeme predstaviť aj široký komplex strojného, manipulačného, transportného, meracieho, kontrolného a pomocného zariadenia, ktoré je počas výroby súčiastky riadené počítačom. CAM ako software slúži na prípravu CAD dát vo výrobnom procese a na uskutočnenie simulácií výrobného procesu. Na obrázku 2.3 je príklad CAM systému, konkrétne sa jedná o EdgeCAM.

Priemyselnú výrobnú aktivitu môžeme vo všeobecnosti rozdeliť do 3 kategórií:

1. Masová výroba (automobily)

2. Hromadná výroba (priemyselné stroje, lietadlá)
3. Malovýroba, často výroba 1 kusu (prototypy)

CAM nachádza uplatnenie vo všetkých odvetviach výroby. Predovšetkým dnešný trend veľkého množstva variácií a malej životnosti produktov vyžaduje, aby bol čas potrebný pre prípravu výroby čo najmenší. Jediný spôsob, ako to zaistiť, je vylepšenie výrobných metód a CAM je v tomto ohľade nevyhnutnosťou.



Obrázok 2.3: EdgeCAM

Výhody využívania CAM

Väčšia sloboda pri návrhu – všetky potrebné zmeny návrhu môžu byť vykonávané v ľubovoľnej fáze návrhu.

Zvýšenie produktivity – fakt, že všetka výrobná aktivita je kompletne organizovaná počítačom, umožňuje zvyšovať produktivitu.

Flexibilita obsluhy – CAM zvyšuje flexibilitu výrobných metód a zmien produktovej rady.

Čas potrebný na prípravu výroby je výrazne zredukovaný.

Vyššia spoľahlivosť – vzhľadom na lepšie metódy a riadenie v štádiu výroby, vyrábaný produkt, rovnako ako výrobný systém, sú veľmi spoľahlivé.

Zníženie údržby – väčšina komponent CAM systému obsahuje vstavané diagnostické a monitorovacie zariadenia, takže nevyžadujú takú údržbu, ako konvenčné metódy.

Zníženie odpadu a potreby prerábania – keďže sú pri výrobe používané CNC stroje a využívajú sa štandardné komponenty, tak úroveň vzniknutého odpadu je nízka a nutnosť prerábania je veľmi nízka, ak vôbec je.

Zlepšenie kontroly riadenia vďaka použitiu počítačov.

CAD/CAM

Počítačom podporovaný systém s integrovanou podporou návrhu a súčasne aj výroby súčiastky. Zriedkavo sa používa aj skratka CADM (Computer Aided Design and Manufacturing). Postupne s vývojom CAD, CAPP a CAM systémov sa začali vyvíjať systémy, ktoré by integrovali túto reťaz činnosti – návrh súčiastky – návrh technológie – výroba, do jedného systému. Systémy CAD/CAM integrujú modelovanie súčiastky a jeho konštrukčný návrh, návrh technologickej dokumentácie vo forme NC programov a operatívne riadenie výroby do jedného počítačového systému. CAD/CAM systémy sú v súčasnosti veľmi populárne a prežívajú svoj „boom“. Výhodou týchto integrovaných systémov je ich schopnosť riešiť komplexné a zložité úlohy. Integrujúcou časťou CAD/CAM systémov je model výrobku a vnútorná databáza systému.

2.3.2 CAE

Počítačom podporované inžinierstvo – tento pojem je skôr známy ako automatizácia inžinierskych prác (AIP), resp. počítačom podporované inžinierske práce. Jedná sa napr. o súbor inžinierskych činností, ktoré sa uskutočňujú v predvýrobných etapách realizácie súčiastky. Často sa týmto pojmom komplexne označuje počítačová podpora inžinierskych prác v predvýrobných etapách a zahŕňa systémy ako CAD, CAD/CAM a CAPP. Myslia sa tým všetky činnosti vo vývojovo-návrhovej etape, keď je súčiastka podrobovaná mnohým analýzám, testujú sa rôzne varianty, optimalizuje sa návrh a pod.. Je to komplexné pomenovanie počítačovej podpory návrhovej etapy počas realizácie súčiastky.

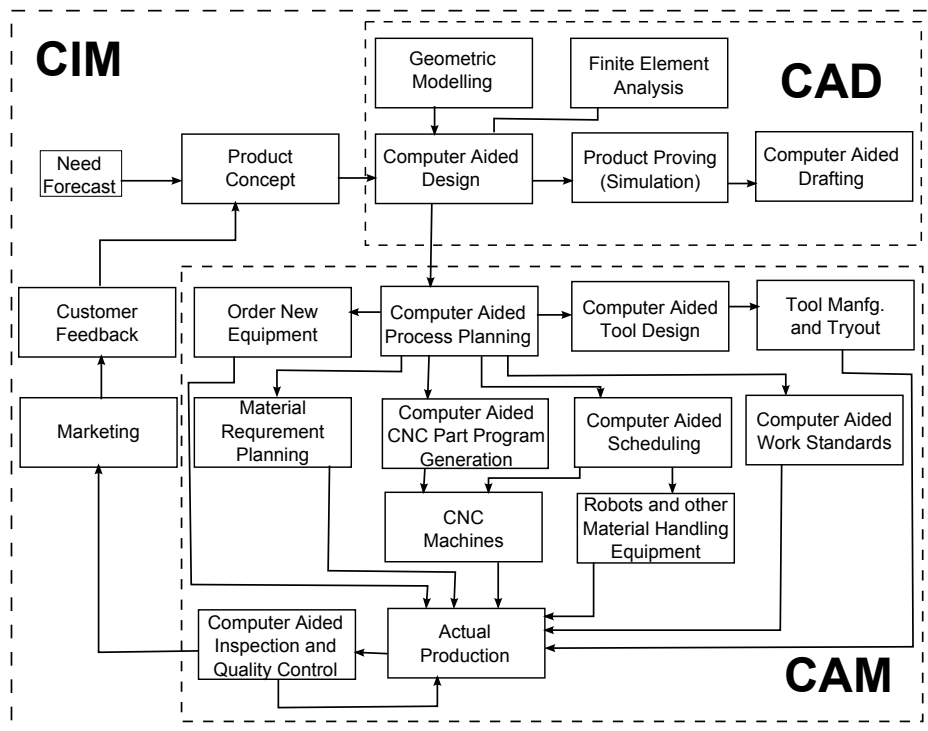
2.3.3 CAPP

Reprezentuje počítačovú podporu pri návrhu a tvorbe technologickej dokumentácie. Ich hlavné uplatnenie je v strojárstve, kde na základe konštrukčnej dokumentácie pomáhajú navrhovať a vytvárať technologickú dokumentáciu. V technologickej dokumentácii sú informácie, ktoré určujú čo sa ide vyrábať, ako (aké výrobné metódy), kde (na akom strojnom zariadení), čím (akými nástrojmi, pri akom upnutí a pod.) a za akých podmienok (technologické podmienky). CAPP systémy umožňujú vytvárať rôzne formy technologickej dokumentácie (slovné, obrázkové technologické postupy, NC programy a pod.). NC programy sú určené väčšinou pre rôzne typy číslicovo riadených strojov (obrábacie, tvárniace stroje, drôtové rezačky a pod.) ako i pre priemyselné roboty.

Výstupom CAPP systémov je technologická dokumentácia, ktorá slúži okrem samotnej výroby aj pre operatívne plánovanie a riadenie výroby. CAPP systémy sú väčšinou vytvárané v databázových systémoch a tvoria dôležité prepojenie medzi CAD a CAM systémami.

2.3.4 CIM

Computer Integrated Manufacturing – nepredstavuje systém, ale integráciu systémov, zúčastňujúcich sa priamo alebo nepriamo na realizácii výrobku. CIM teda môžeme vnímať ako komplex navzájom integrovaných systémov, ktoré sa zúčastňujú na realizácii výrobku. Je to pohľad na integráciu počítačom podporovaných systémov v predvýrobných a výrobných etapách. Ako základ pre širšiu integráciu CA systémov v podniku sa uvažujú systémy CAD, CAD/CAM, CAM, CAPP systémy a iné.



Obrázok 2.4: Produkčný cyklus v prostredí počítačovej výroby [11]

2.4 Požiadavky na CAD systém

CAD systémy sú v dnešnej dobe používané ako skutočné konštrukčné nástroje, ktoré musia spĺňať patričné požiadavky.

2.4.1 Presnosť

- Numerická presnosť, teda aby súradnice a rozmery objektov boli v desatinných číslach
- Presný popis 2D/3D geometrie, teda presnosť plôch a kriviek
- Presné geometrické operácie a analytické riešenie geometrických úloh – výpočet priesečníkov, piat kolmíc, bodov dotykov, zistenie veľkosti plochy, uhlu, a pod.
- Možnosť kresliť objekty viacerými spôsobmi a presne zadávať ich súradnice
- Možnosť vykonávať presné operácie – vykonávať transformácie objektov, uchopovať body k iným významným bodom.

Kreslenie objektov

Zo všeobecnejšieho hľadiska môžeme kreslenie rozdeliť na kreslenie:

- myšou – je rýchlejšie, dynamickejšie, no nemusí byť príliš presné
- klávesnicou – pomalšie, no keďže sú parametre objektov zadávané ručne z klávesnice, je oveľa presnejšie

CAD systémy obvykle ponúkajú viacero spôsobov ako jednotlivé konkrétne entity kresliť:

- Úsečky – najbežnejší spôsob kreslenia úsečky je zadaním jej krajných bodov. Ďalšími možnosťami sú zadanie stredu (alebo počiatočného bodu), dĺžky a odchýlky od osi x . Špeciálne metódy nám umožnia úsečku kresliť len v horizontálnej alebo vertikálnej polohe, ako dotyčnicu k iným objektom alebo ako kolmicu na iné objekty (dotyčnice a kolmice sú vlastne špeciálnym prípadom uchopovania, kedy sa uchopuje k bodom, ktoré musia byť dynamicky dopočítavané), prípadne vytvárať lomenú čiaru.
- Kružnice – základom pre vykreslenie kružnice sú jej stred a polomer. Tiež je možné zadať stred a bod na kružnici a dopočítať polomer alebo 2 body na kružnici tvoriace jej priemer a dopočítať polomer a súradnice stredu. Zložitejšou alternatívou je zadanie 3 bodov ležiacich na kružnici.
- Elipsy – elipsu možno štandardne určiť jej stredom a veľkosťou poloôs. Inou možnosťou je určenie 2 protiľahlých vrcholov obdĺžnika, ktorému je elipsa vpísaná. Zo súradníc vrcholov je vypočítaný stred elipsy a dĺžky jej poloôs budú rovné polovičným dĺžkam strán obdĺžnika.

Transformácie objektov

Vzhľad vykreslených objektov by nemal byť konečný a mal by byť upraviteľný podľa potrieb užívateľa systému. Na to slúžia transformácie objektov a s nimi súvisiaci výber objektov. Základné transformácie objektov sú posun, rotácia, zmena veľkosti, skosenie, zrkadlenie.

Za transformácie možno považovať tiež zmenu veľkosti a posunutie metódou „potiahnuť a pustiť“, pričom ťahať sa dajú významné body objektov (vrcholy, krajné body, kvadrantové body, ...) alebo celé objekty. Špeciálnym prípadom je napr. ťahanie vrcholu obdĺžnika s tým, že bude zmenená poloha len tohoto bodu a všetky ostatné svoju polohu nezmenia.

Uchopovanie

Spomenuli sme, že kreslenie myšou nie je príliš presné. Uchopovanie je spôsob, ako tento spôsob kreslenia spresniť. Jedná sa o metódu „prichytávania“ významných bodov objektov k významným bodom iných objektov. Bod je uchopený k inému, keď je k nemu dostatočne blízko, resp. keď je kurzor dostatočne blízko. Tým je odstránená nutnosť ručne prepočítavať parametre, čo prácu zjednodušuje a najmä zrýchľuje. Uchopovanie funguje nielen k významným bodom objektov, ale aj iným dôležitým bodom scény. Bodmi na uchopenie môžu byť:

- Vrcholy polygónov a krajné body úsečiek
- stredy objektov, prípadne stredy strán polygónov
- kvadrantové body kružníc a elíps
- priesečníky objektov
- body mriežky

2.4.2 Organizácia

Hladiny – dávajú nám možnosť pracovať s návrhom nielen ako s celkom, ale aj ako s modelom, ktorý je štruktúrovaný a možno izolovane meniť vlastnosti jeho častí. Hladiny sú akoby priehľadné fólie, ktoré sú preložené cez seba, definujú rôzne vlastnosti entít danej hladiny, napr. farbu, typ a hrúbku čiary, atď. a je možné ich skrývať alebo zobrazovať. Zmena viditeľnosti má veľký význam napríklad pri rozsiahlych návrhoch, kedy v záujme zvýšenia prehľadnosti môžeme skryť hladiny, ktoré v danom momente nie sú potrebné. Entity sú pomocou hladín logicky rozčlenené do skupín podľa spoločných vlastností.

2.4.3 Reprezentácia dát v 2D CAD systéme

V CAD systéme pre 2D kreslenie sú dáta popísané vektorovými entitami 2D grafiky (úsečky, kružnice, oblúky, krivky, plochy, ...). Tieto entity musia spĺňať požiadavky presnosti a organizácie, musí byť možná ich editácia (orezanie, zaoblenie, ...) a transformácia (posunutie, zrkadlenie, kopírovanie, ...).

CAD systémy s rovnakou reprezentáciou môžu teoreticky zdieľať dáta, ktoré sú prioritne vektorové. Zdieľanie je umožnené vďaka predpokladu, že systémy zhodne rozumejú a zobrazujú základné entity vektorovej grafiky.

Vektorová grafika je spôsob popisu (uloženie, definícia) spracováanej a zobrazovanej informácie vo forme skupiny vektorových entít (úsečky, kružnice, krivky, polygóny, ...), analyticky. Tento popis dát získame buď manuálne, alebo syntézou (generovaním alebo prevodom z iného popisu). Všetky objekty definované vo vektorovej grafike sú tvorené zoznamom základných entít. Uchováваме teda informáciu o popise zobrazených objektov. Pri každom zobrazení sa popis prevádza na rastrové zobrazenie. Jednotlivé objekty a ich entity stále existujú, a preto môžeme podľa potreby meniť ich vlastnosti (farba, štýl, ...) a parametre (polohu, veľkosť, orientáciu, ...). Jednotlivé entity sú definované presne matematicky (geometria entít, súradnice bodu, ...) a parametricky (farby, štýl čiar, ...). Entity sú teda popísané nezávisle na rozlíšení ich zobrazenia, na rozdiel od zobrazenia v rastrovej grafike. Keď sa pozrieme na detail zobrazenia vektorových objektov, je ich zobrazenie prevedené na aktuálne výstupné rastrové rozlíšenie s maximálnou úrovňou detailu.

2.4.4 Prenositelnosť výkresov medzi rôznymi CAD systémami

Data eXchange File je formát vyvinutý firmou Autodesk pre prevod 2D výkresov. Pomocou DXF je možné preniesť aj 3D drôtové modely. Vďaka svojej jednoduchej štruktúre dosiahol tento formát veľké rozšírenie a dnes je podporovaný takmer každým CAD systémom. S formátom DXF dokážu pracovať i iné programy ako CAD systémy alebo konverzné utility pre CAD. Formát DXF existuje v množstve verzií, najčastejšie sa stretávame s verziou DXF odpovedajúcou exportu z AutoCADu R12.

Formát DWG je interný binárny formát programov od najväčšieho výrobcu programov CAD, a to spoločnosti Autodesk. Podobne ako u formátov DXF sa tento formát stal štandardom pre rysovanie počítačových projektov v najrôznejších oboroch. Ide o nezverejnený vektorový formát, ktorý sa vyvíjal s každou ďalšou verziou. Existujú rôzne konvertory medzi verziami formátu DWG. Dáta uložené vo formáte DWG je možné otvoriť napríklad v programe Corel Draw, alebo Adobe Illustrator. Pre výmenu dát s inými aplikáciami by ale mal slúžiť formát DXF.

2.4.5 Uživateľské rozhranie v CAD systémoch

CAD systémy predstavujú veľký pokrok v konštruovaní technických výkresov, keďže ich nie je nutné tvoriť ručne. Stále je však potrebný ľudský faktor v konštruovaní výkresov, počítače to neurobia samé za nás. K tomu, aby človek mohol vytvoriť výkres pomocou CAD systému, potrebuje tento systém grafické uživateľské rozhranie (GUI). Rozhranie by malo byť efektívne navrhnuté a dostatočne rýchle, keďže cieľom CAD systémov je rýchlejšie a efektívnejšie vytváranie výkresov. Jednoduché, intuitívne, prehľadné ovládanie sú kľúčom k tomu, aby si užívateľ na ovládanie rýchlo zvykol, pamätal si štruktúru ovládacích prvkov rozhrania, a aby sa mu pracovalo s rozhraním čo najpohodľnejšie. Nemenej dôležitá je celková vizuálna stránka rozhrania, ktorá zanechá v užívateľovi dojem a je už na tvorcovi rozhrania, či bude tento dojem pozitívny alebo negatívny. Cieľom je, pochopiteľne, pozitívny dojem. Štandardnými prostriedkami na ovládanie CAD systému sú:

- myš – výber ikon, ovládanie menu a jeho ponúk, výber objektov na obrazovke, pričom tlačidlá myši môžu byť konfigurovateľné
- klávesnica – využívanie klávesových skratiek, zadávanie parametrov funkcií a používanie príkazového riadka (príkazy spracováva skriptovací jazyk)
- tablet, pilot, navigátor – výrazne znižujú čas potrebný na prácu

Kapitola 3

Návrh a implementácia aplikácie

V tejto kapitole si povieme o cieľoch, ktoré chceme dosiahnuť, návrhu aplikácie, jej implementácii, teda o spôsobe, ako dospieť k stanoveným cieľom, o jej štruktúre a o nástrojoch a prostriedkoch využitých pri tvorbe. Budú uvedené základné časti aplikácie:

- reprezentácia dát a spôsob ich zobrazovania
- dôležité funkcie aplikácie
- grafické užívateľské rozhranie

Po prečítaní kapitoly bude čitateľ vedieť z akých častí aplikácia pozostáva, čo majú jednotlivé časti na starosti a bude oboznámený so základnými prvkami užívateľského rozhrania. Rozbor, ako aplikácia pracuje a ako vykonáva svoje funkcie, bude rozoberaný postupne s návrhom.

3.1 Ciele

Základom pre vytváranie aplikácie bolo stanovenie cieľov, teda jej možností a funkcií, ktoré budú v konečnom štádiu k dispozícii. Istý náčrt cieľov bol spomenutý už v predošlej kapitole v sekciách venovaných požiadavkam na CAD systém 2.4, reprezentácii dát 2.4.3 a grafickému užívateľskému rozhraniu 2.4.5. Tento náčrt viac špecifikujeme a skonkretizujeme pre navrhovanú aplikáciu:

- Stanovenie grafických entít, ktoré bude systém schopný kresliť – úsečka, obdĺžnik, kružnica, elipsa, krivka.
- Kreslenie myšou 2 základnými spôsobmi:
 - stlačenie tlačítka v počiatočnom bode → posun kurzoru so stlačeným tlačítkom → uvoľnenie tlačítka v koncovom bode
 - prvé kliknutie tlačítka v počiatočnom bode → posun kurzoru s uvoľneným tlačítkom → druhé kliknutie tlačítka v koncovom bode
- Zadávanie parametrov entity, ktorú chceme vykresliť, z klávesnice.
- Možnosť zadávať relatívne a polárne súradnice (polárne len v prípade úsečky).

- Hladiny zjednocujúce určité entity do skupín, s možnosťou meniť farbu a viditeľnosť entít v hladine, pridávať a odstraňovať hladiny.
- Transformácie celého plátna – posun a priblíženie/oddialenie plátna.
- Výber entít – obdĺžnikovým výberom alebo kliknutím myšou na entitu.
- Transformácie vybraných entít – posun, kopírovanie na určitú pozíciu, zmena veľkosti a rotácia. V prípade rotácie mať možnosť zadať bod, okolo ktorého budú vybrané entity rotovať.
- Posun vybranej entity ako celku alebo jedného z jej bodov pomocou myši.
- Ortogonálny mód kreslenia úsečky – horizontálne alebo vertikálne.
- Možnosť vykresľovať úsečku ako kolmicu a dotyčnicu.
- Uchopovací mód – uchopovanie k významným bodom entít.
- Možnosť uložiť prácu do súboru a načítať ju zo súboru.
- Prehľadné, intuitívne a jednoducho ovládateľné užívateľské rozhranie.

3.2 Dátová reprezentácia

V predchádzajúcej kapitole sme si povedali, že CAD systémy pracujú s vektorovou grafikou, pričom základné vektorové entity, s ktorými navrhnutá aplikácia pracuje sú úsečka, obdĺžnik, kružnica, elipsa a krivka. Všetky spomenuté entity sú dané bodmi, prípadne vzdialenosťami (polomer kružnice, poloosi elipsy) vyjadrenými v desatinných číslach, čo nám umožňuje vykonávať nad entitami presné geometrické operácie, ktoré sú pre správne fungovanie aplikácie nevyhnutné.

3.2.1 Entity

Každá entita je reprezentovaná vlastnou triedou, ktorá je potomkom abstraktnej triedy **Shape**. Sú to triedy **Line**, **Rectang**, **Circle**, **Ellip** a **Curve**. Trieda **Shape** obsahuje atribúty a metódy spoločné pre všetky objekty. To nám umožňuje v určitých situáciách pristupovať k entitám jednotným spôsobom a taktiež nám to umožňuje jednoducho rozšíriť aplikáciu o nové typy entít. Vlastné triedy odvodené od triedy **Shape** obsahujú už atribúty a metódy špecifické pre daný typ entity. Všetky body sú objektmi triedy **QPointF**, ktorej atribúty reprezentujúce súradnice sú typu **qreal**. Tento typ je alternatívou typu **double**. Rovnako všetky dĺžky, vzdialenosti a výsledky operácií sú typu **qreal**, ktorý zabezpečuje dostatočný rozsah a presnosť.

Najdôležitejším atribútom triedy **Shape** je identifikátor **ID**. Dátový typ použitý pre identifikátor je **quint64**, ktorý je ekvivalentom typu **unsigned long long int**. Novovytvorenému objektu musí byť v rámci plátna priradený jednoznačný identifikátor v tvare **0|LLL|INDEX** (znak **|** je len pre ilustráciu, ako oddeľovač), kde **0** je číslo udávajúce typ objektu (1 úsečka, 2 obdĺžnik, 3 kružnica, 4 elipsa, 5 krivka), **LLL** je trojčiferné číslo určujúce index hladiny, ktorá je práve aktívna a **INDEX** je index objektu v zozname. Počet čísel **INDEXu** je ohraničený len rozsahom použitého dátového typu. Takto vytvorený identifikátor je pre každý objekt jedinečný a ďalší význam tohoto spôsobu identifikácie si priblížime

neskôr v súvislosti s uchopovacím a výberovým módom. Zvyšnými atribútmi sú farba objektu, premenná indikujúca, či je objekt zvýraznený výberovým módom, typ čiary a stred objektu.

Zo spoločných metód je navýznamnejšou **draw**, ktorá pomocou OpenGL vykreslí konkrétny objekt na plátno. Funkcia **highlight** slúži na zvýraznenie objektu, keď bol vybraný užívateľom. Dôležité body objektu sú vyznačené pomocou malých bielych štvorčiek, ktorých vykreslenie zaisťuje funkcia **glRectf**. Dvojica **serialize** a **deserialize** je používaná pri uložení a načítaní dát zo/do súboru. Ich úlohy sú dáta serializovať a deserializovať do/z binárneho dátového toku. Konkrétnejší popis uvádza sekcia 3.4.6.

Špecifikácia jednotlivých entít:

Úsečka – určená 2 bodmi, metódou **setMiddle** je vypočítaný jej stred.

Obdĺžnik – tvorený 4 vrcholmi, pri kreslení stačí zadať 2 protilahlé, zvyšné sú dopočítané funkciou **setCorners**. Stredy strán sú vypočítané funkciou **setSidePoints** a sú uložené v poli **sidePoints**. Obdĺžnik si uchováva aj uhol jeho rotácie v stupňoch, je to celé číslo v rozsahu 0–359. Metódou **draw** je vykreslený ako 4 úsečky.

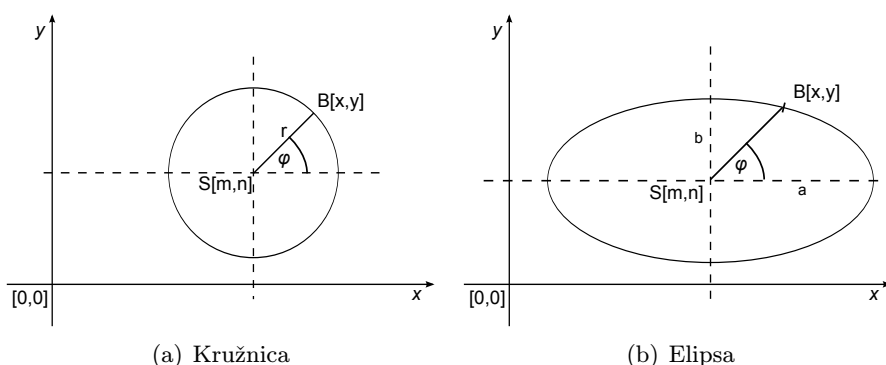
Kružnica – daná stredom a polomerom. Kvadrantové body sú počítané metódou **setPoints** a ukladá ich do poľa. Algoritmus pre vykreslenie kružnice v metóde **draw** pracuje tak, že ju kreslí ako mnohouholník, ktorého stred je identický so stredom kružnice. Súradnice bodu *B* na obrázku 3.1(a) sú počítané vzťahmi:

$$x = r \cdot \cos \varphi + m \quad y = r \cdot \sin \varphi + n$$

Body mnohouholníka sú týmto spôsobom počítané pre $\varphi \in [0, 2\pi]$ s krokom $\frac{\pi}{180}$ rad. Medzi jednotlivými bodmi sú vykreslené úsečky. **Elipsa** – daná stredom a poloosami. Metódy **setPoints** a **setCorners** počítajú kvadrantové body a vrcholy obdĺžnika, ktorému je elipsa vpísaná. Rovnako ako obdĺžnik, aj elipsa si uchováva uhol rotácie. Algoritmus pre vykreslenie elipsy v metóde **draw** pracuje tak, že ju kreslí ako mnohouholník, ktorého stred je identický so stredom elipsy. Súradnice bodu *B* na obrázku 3.1(b) sú počítané vzťahmi:

$$x = b \cdot \cos \varphi + m \quad y = a \cdot \sin \varphi + n$$

Body mnohouholníka sú týmto spôsobom počítané pre $\varphi \in [0, 2\pi]$ s krokom $\frac{\pi}{180}$ rad. Medzi jednotlivými bodmi sú vykreslené úsečky.



Krivka – Beziérová krivka, zadávaná je pomocou kontrolných bodov uložených v zozname typu **QVector**. Metóda **generateCurve** vypočíta 50 bodov, ktoré sú spojené úsečkami

a tým vznikne krivka. Pri výpočtoch využíva funkciu `bern` na vyčíslenie Bernstenovho polynómu. Pri veľkom množstve kontrolných bodov sa krivka môže na určitých úsekoch javiť ako lomená čiara. Tento jav by sa dal odstrániť zvýšením počtu bodov krivky, ale len za cenu väčšej výpočetnej náročnosti.

Alternatívne spôsoby kreslenia

Rôzne situácie si vyžadujú rôzne spôsoby zadávania objektov, preto ich bolo v aplikácii navrhnutých hneď niekoľko. Povieme si ako o spôsoboch zadávania pomocou myši a klávesnice, tak o 2 možnostiach zadávania konkrétnych entít, ktorými sú kružnica a elipsa.

Objekty oboch entít môžu byť kreslené 2 spôsobmi. Kružnica sa dá určiť zadaním stredu a polomeru alebo 2 bodov ležiacich na kružnici, ktorých spojením dostaneme úsečku prechádzajúcu stredom kružnice, priemer. Ďalšou alternatívou zadávania kružnice by mohlo byť zadanie 3 bodov ležiacich na kružnici. Elipsu možno určiť zadaním stredu a poloosi alebo 2 protiľahlých bodov obdĺžnika, ktorému bude výsledná elipsa vpísaná. Spojením týchto bodov dostaneme uhlopriečku daného obdĺžnika.

Spôsoby zadávania objektov pomocou myši a klávesnice sú uvedené v tabuľke 3.1.

	úsečka	obdĺžnik	kružnica	elipsa	krivka
myš	áno	áno	áno	áno	áno
klávesnica	áno	áno	áno	áno	nie
kombinovane	áno	nie	áno	áno	nie

Tabuľka 3.1: Tabuľka znázorňuje spôsoby vykresľovania jednotlivých objektov

Ako sme si mohli všimnúť, krivku je možné zadať len pomocou myši. Rozšírenie o zadávanie krivky klávesnicou súvisí so zovšeobecnením zadávania všetkých parametrov pomocou klávesnice. Toto zovšeobecnenie bude uvedené v sekcii 3.5 venovanej návrhu grafického užívateľského rozhrania.

Objekty, u ktorých je uvedená možnosť kombinovaného kreslenia, majú túto možnosť len v určitých prípadoch:

- úsečka – pri zadávaní polárnych súradníc, klávesnicou určíme dĺžku a odchýlku úsečky, kliknutím na plátno určíme počiatočný bod
- kružnica – pri zadávaní stredu a polomeru, klávesnicou určíme polomer, kliknutím na plátno určíme stred
- elipsa – pri zadávaní stredu a poloosi, klávesnicou určíme poloosi, kliknutím na plátno určíme stred

Ak pri zadávaní použijeme len myš, máme opäť 2 možnosti ako s myšou pracovať. Buď stlačíme tlačítko, čím zadáme počiatočný bod, posúvame myš so stlačeným tlačítkom a uvoľníme ho v koncovom bode alebo len klikneme v počiatočnom bode, posúvame myš s uvoľneným tlačítkom a klikneme druhýkrát v koncovom bode.

3.2.2 Dátové kontajnery – hladiny a plátna

Hladiny – trieda Layer

Jednou z požiadaviek na výslednú aplikáciu bola organizácia objektov do hladín. Hladiny nám umožňujú pracovať s návrhom ako s komplexným celkom zloženým z viacerých izolovaných častí, ktoré môžeme nezávisle na sebe modifikovať.

Hladina je reprezentovaná triedou **Layer**, ktorej základné atribúty sú názov, farba jej prislúchajúcich objektov, viditeľnosť a zoznamy jednotlivých objektov podľa ich typov. Každý zoznam je objektom triedy **QVector**. Po zadaní parametrov kresleného objektu je metódou **addObject** objekt vytvorený a pridaný do príslušného zoznamu. Trieda **Layer** tiež implementuje metódy na mazanie objektov. Mazať sa dajú len vyznačené objekty a to klávesou *Del*. Po jej stlačení je zavolaná funkcia **deleteSelectedObjects**, ktorá postupne vyznačené objekty maže metódou **deleteObject**. Ďalšou metódou na mazanie objektov je **deleteObjects**, ktorá zmaže všetky objekty hladiny. Na zmenu farby slúži funkcia **colorChange**.

Aktívna je vždy len 1 hladina a každý vykreslený objekt bude vždy patriť len do 1, a to práve aktívnej hladiny. Implicitne je aktívna hladina s názvom **default** a je to jediná hladina, ktorá sa nedá zmazať. Musí teda existovať aspoň 1 hladina, objekty nemôžu byť vykresľované samostatne, bez príslušnosti k hladine. Užívateľovi je umožnené hladiny pridávať, mazať, vyberať aktívnu hladinu, meniť ich farby a viditeľnosť. V návrhu nie je braná do úvahy možnosť, že by mal objekt inú farbu a viditeľnosť ako hladina, do ktorej patrí.

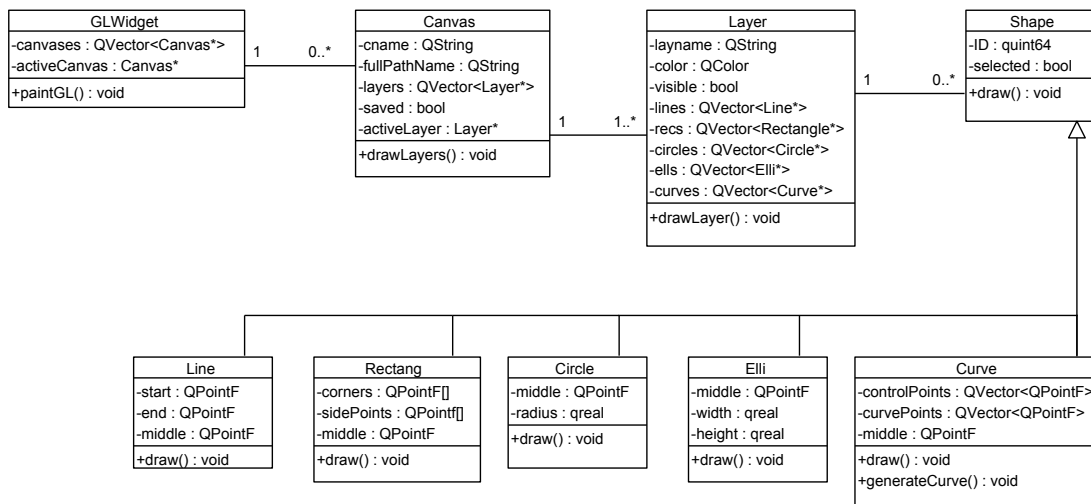
Plátna – trieda Canvas

S čím však návrh počíta, je členenie na celky združujúce určité skupiny hladín – plátna. Plátno je reprezentované triedou **Canvas**, ktorej najdôležitejšími atribútmi sú názov plátna, cesta k súboru, v ktorom sú uložené dáta všetkých hladín plátna a ich objektov, zoznam hladín príslušných danému plátnu a ukazovateľ na aktívnu hladinu. Nemenej dôležité sú aj mnohé ďalšie atribúty, napr. ukazovatele na objekty tried, ktoré slúžia na prácu s entitami na plátne a plátnom samotným, alebo aj atribúty určujúce aktívny mód. Metódy plátna slúžia najmä na prácu s hladinami a ich objektmi. Pridanie hladiny vykonáva funkcia **createLayer** a mazanie majú na starosti metódy **deleteLayer** a **deleteLayers**. Druhá menovaná zmaže z plátna všetky hladiny a je volaná len pri jeho zatvorení. Metódy na transformácie objektov budú popísané v sekcii 3.4.4. Trieda poskytuje tiež funkcie, ktoré pomáhajú dopočítavať parametre vytváraných objektov, konkrétne polomer, poloosi a stred kužeľosečiek.

Rovnako ako u hladín, aj plátno môže byť aktívne práve 1, objekty nakreslené na scénu budú patriť aktívnemu plátnu. Užívateľ má taktiež možnosť zatvoriť všetky plátna, čím sa aplikácia dostane do neaktívneho stavu, v ktorom zotrva až do vytvorenia nového plátna, resp. otvorenia súboru s uloženými dátami. V tomto stave nie je možné na scénu nič kresliť a celé užívateľské rozhranie pre prácu so scénou je deaktivované.

Obe triedy implementujú taktiež metódy **serialize** a **deserialize**, ktoré serializujú a deserializujú informácie o plátne, resp. hladine.

Objektový návrh organizácie dát je znázornený na čiastočnom diagrame tried na obrázku 3.1.



Obrázok 3.1: Zjednotený diagram tried pre znázornenie organizácie dát

3.3 Spôsob zobrazovania

Objekty je nutné nielen vhodne reprezentovať a zorganizovať, ale aj vhodne vizualizovať pre užívateľa. Na zobrazenie objektov aktívneho plátna slúži čierna plocha – scéna, ktorá je najdôležitejšou časťou grafického užívateľského rozhrania. Jedná sa v podstate o objekt triedy `GLWidget` zdedenej z triedy `QGLWidget`, ktorá predstavuje OpenGL nadstavbu pre triedu `QWidget`. Obsahuje zoznam plátien a ukazovateľ na aktívne plátno, čím získava prístup k metódam pracujúcim s plátnom, jeho hladinami a ich objektmi. Jednotka dĺžky je pri pôvodnom zobrazení 10 pixelov. V budúcich rozšíreniach aplikácie by mohli byť brané do úvahy reálne jednotky, ako napr. centimetre alebo milimetre. Počiatok súradnicovej sústavy sa nachádza v ľavom dolnom rohu scény.

Metódy scény

Scéna predstavuje základný prvok užívateľskej interakcie s aplikáciou. Jej hlavnými úlohami je správne zobrazovať nakreslené entity a reagovať na podnety z okolia.

Základ triedy `GLWidget` tvoria 3 virtuálne metódy:

- **initializeGL** – táto metóda je volaná len 1 raz a to pri vytvorení scény pri spustení aplikácie. Jej úlohou je počiatočná inicializácia scény.
- **paintGL** – spôsobí prekreslenie scény, teda všetkých viditeľných hladín aktívneho plátna a tiež má na starosti volanie transformácií plátna a prípadné vykreslenie stredu rotácie. Metódu je možné explicitne zavolať príkazom `updateGL`.
- **resizeGL** – metóda je volaná pri každej zmene rozmerov scény a jej parametrami sú aktuálne nové rozmery. Zabezpečuje prispôbovanie súradnicovej sústavy novým rozmerom, zmena nebude mať vplyv na jednotku dĺžky a pozíciu stredu sústavy.

Aby užívateľ mohol so scénou pohodlne pracovať, trieda implementuje reakcie na jeho akcie, najmä s myšou, ktorá je pre kreslenie objektov a transformácie scény nevyhnutná. Jedná sa o stlačenie a uvoľnenie tlačítka, pohyb s uvoľneným alebo stlačeným tlačítkom

a točenie kolieska myši. Jedinou reakciou scény na koliesko je približovanie a oddaľovanie plátna. Pri každej akcii (okrem kolieska) sú prepočítané súradnice kurzoru na základe aktuálneho priblíženia a posunutia scény, sú použité podľa aktívneho módu aplikácie a tlačítka myši a sú poslané signálom hlavnému oknu, aby ich aplikácia mohla poskytnúť užívateľovi. Obsluha akcií tiež rozoznáva či sú objekty kreslené ťahom alebo klikaním a pamätá si body, v ktorých boli tlačítka stlačené a uvoľnené.

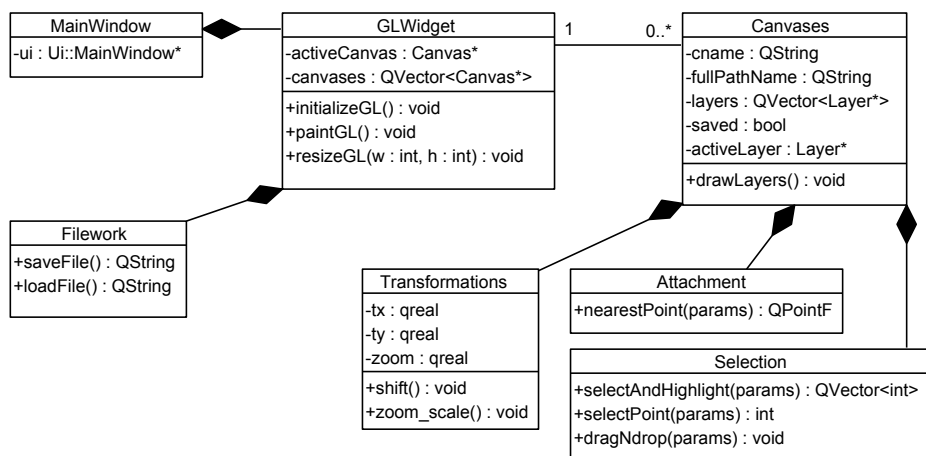
Metóda reagujúca na pohyb myši musí rozoznávať či je pri pohybe stlačené tlačítko, prípadne ktoré tlačítko. Ak ľavé, tak sa jedná buď o kreslenie alebo výber. Ak pravé, tak užívateľ mení polohu vybraného bodu niektorého z objektov. V situácii, keď nie je stlačené žiadne tlačítko a je aktívne kreslenie a uchopovanie zároveň, tak aplikácia zvýrazní najbližšie body vyhovujúce uchopeniu a prípadne dynamicky kreslí objekt zadávaný 2 kliknutiami.

Pri uvoľnení tlačítka sa na základe módu uloží nový objekt alebo sa vyberú vyhovujúce objekty. Špeciálnym prípadom je uvoľnenie pravého tlačítka pri zadávaní krivky. Ak boli zadané aspoň 3 kontrolné body, tak krivku uloží, inak zruší jej kreslenie.

Dôležitou je tiež funkcia na vytvorenie nového plátna.

3.4 Významné funkcie aplikácie

CAD systém nie je len o kreslení základných tvarov, ale najmä o vytváraní niečoho konkrétneho, presného, prehľadného a najmä flexibilného. Asi je len ťažko predstaviteľné, že nakreslíme úsečku a po zistení, že sme ju nenakreslili tak presne, ako potrebujeme, sme nútení ju zmazať a kresliť až kým sa „netrafíme“. Preto CAD systém disponuje celým radom možností a funkcií ako s jeho pomocou spresniť a urýchliť kreslenie. V tejto sekcii budú popísané najvýznamnejšie funkcie navrhovanej aplikácie. Zjednodušený objektový návrh tried, ktoré ich implementujú je na obrázku 3.2.



Obrázok 3.2: Zjednodušený diagram tried pre znázornenie tried implementujúcich dôležité funkcie aplikácie

3.4.1 Transformácie plátna

V aplikácii sa zobrazuje len určitý výsek plátna a bolo by veľmi nepraktické, ak by mal užívateľ na kreslenie k dispozícii iba tento výsek. Z toho dôvodu je možné sa na plátne

posúvať a meniť jeho priblíženie, ktorého hranice sú 0–999%.

Posúvať plátno je možné pomocou kombinácie *Ctrl* + ťahanie plochy pomocou ľavého tlačítka myši.

Približovať, resp. oddaľovať plátno možno pomocou tlačítiek v užívateľskom rozhraní, použitím kolieska myši, keď sa kurzor nachádza nad plátnom (od seba pre priblíženie, k sebe pre oddialenie) alebo pomocou klávesových skratiek *Ctrl* + + pre priblíženie a *Ctrl* + – pre oddialenie. Pôvodné priblíženie dosiahneme stlačením príslušného tlačítka v užívateľskom rozhraní alebo klávesovou skratkou *Ctrl* + 0. Jeden krok zmeny priblíženia tlačítkom alebo klávesami je rovný 5%. Do textového políčka zobrazujúceho aktuálne priblíženie môže užívateľ napísať hodnotu v danom rozsahu a potvrdiť ju klávesou *Enter*.

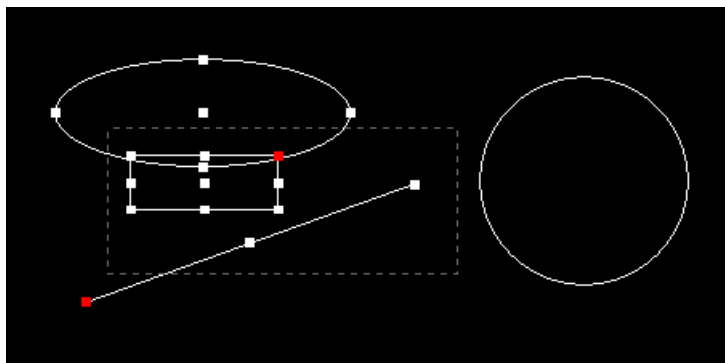
Vykonanie popísaných transformácií majú na starosti metódy triedy **Transformations**:

- **shift** – posun plátna pomocou OpenGL funkcie **glTranslated**
- **zoom_scale** – zmena merítka plátna pomocou OpenGL funkcie **glScaled**

Trieda obsahuje atribúty, ktoré slúžia na uchovanie aktuálnych súradníc vektoru posunutia a hodnotu merítka daného plátna.

3.4.2 Výberový mód

Aby sme mohli modifikovať a transformovať objekty nakreslené na plátne, je žiaduce, aby aplikácia vedela, ktoré objekty to majú byť. Preto návrh obsahuje výberový mód, v ktorom sa na plátno nedá kresliť, ale dajú sa vyberať už nakreslené objekty. Výber je realizovaný len za pomoci myši a to 2 spôsobmi – kliknutím na objekt alebo obdĺžnikovým výberom. Prvým spôsobom je možné vyberať objekty len po jednom, zatiaľ čo obdĺžnikový výber nám umožňuje vybrať všetky viditeľné objekty. Vybraný objekt zvýrazní svoje významné body. Ukážka obdĺžnikového výberu a zvýraznenia bodov je znázornená na obrázku 3.3.



Obrázok 3.3: Obdĺžnikový výber

Čiarkovaný sivý obdĺžnik predstavuje výberový obdĺžnik. Ako vidíme na obrázku, objekty, ktoré sa čiastočne alebo celkom nachádzajú v čiarkovanom obdĺžniku, majú zvýraznené body, sú vybrané. Jediným objektom, ktorý nie je vybraný, je kružnica, pretože nezasahuje do výberového obdĺžnika.

Pri výbere nie je braný ohľad na to, aká hladina je aktívna, ak sa vo výberovom obdĺžniku nachádzajú objekty z rôznych hladín, sú vybrané všetky. Objekt vybraný nebude, len ak hladina, do ktorej, patrí je zneviditeľnená.

V prípade, že sa vo výbere nachádza neželaný objekt, resp. v ňom nejaký chýba, je možné z výberu objekt odstrániť, resp. ho doň pridať. To nám umožňuje kombinácia *Shift* + výber objektu. Kombinácia funguje ako pri výbere kliknutím, tak obdĺžnikom.

Pozíciu zvýrazneného objektu alebo ktoréhokoľvek z jeho zvýraznených bodov možno pomocou myši zmeniť. Keď sa kurzor nachádza nad jedným z bodov, tak stlačením pravého tlačítka dôjde k detekcii tohoto bodu. Potiahnutie myši so stlačeným pravým tlačítkom spôsobí posunutie daného bodu spolu s kurzorom. Takto môžeme meniť veľkosti a pozície jednotlivých objektov.

Vyznačené objekty sa dajú zmazať pomocou klávesy *delete*.

Programová realizácia výberu

Na starosti ju má trieda **Selection**, konkrétne jej metóda **selectAndHighlight**. Tá využíva OpenGL mechanizmus určený presne k riešeniu takéhoto problému – selection buffer. Ten je najprv nutné vytvoriť a následne ho nastaviť ako selection buffer a OpenGL prepnúť do selection módu zavolaním funkcie **glRenderMode(GL_SELECT)**. Následne je vykonaná inicializácia zásobníku, kam sa budú ukladať mená nájdených objektov. Aby systém vedel, kde a čo má hľadať, určíme okolie kurzoru (výber kliknutím) alebo plochu výberového obdĺžnika. Potom zavoláme vykreslenie všetkých hladín, čo nespôsobí prekreslenie scény, ale uloženie ID objektov v zvolenom okolí do selection bufferu. Pri kreslení metóda **draw** zistí, že OpenGL je v stave výberu a pred samotným vykreslením vloží na zásobník ID objektu pomocou funkcie **glPushName**. Po tomto procese môžeme prepnúť späť do stavu kreslenia (prepnutý bude len OpenGL stav, aplikácia zostáva vo výberovom móde) opäť pomocou funkcie **glRenderMode(GL_RENDER)**, s iným parametrom. Pri tomto prepnutí je nutné si uložiť návratovú hodnotu funkcie, ktorá udáva počet „zásahov“ do objektov. Každému zásahu pripadá množina mien, najčastejšie je táto množina jednoprvková (obdĺžnikový výber alebo kliknutím na 1 objekt). Ak však klikneme na priesečník 2 objektov, tak zásah bude zaregistrovaný len 1 no s 2 menami. Za vybraný bude označený iba 1 z nich, aplikácia nijak neošetruje, ktorý to bude, vyberie ten čo má v bufferi nižší index. Pre každé meno sú zavolané metódy triedy **Layer** na nastavenie, výberu a zvýraznenia objektu. V tomto spočíva hlavný význam tvaru ID objektu. Pred volaním metód pre dané meno je z neho získaný index hladiny, ktorej daný objekt patrí. Volané metódy tiež nemusia prehľadávať zoznamy kým narazia na položku s rovnakým ID, namiesto toho si meno rozložia a dostanú informácie, v ktorom zozname a s akým indexom sa objekt nachádza. Nakoniec sú mená vybraných objektov vložené do zoznamu, ktorý tvorí návratovú hodnotu funkcie.

Trieda implementuje aj ďalšie 2 dôležité metódy, ktoré slúžia na zmenu polohy bodov vybraných objektov. Detekciu, ktorý bod bol vybraný vykonáva metóda **selectPoint** rovnakým spôsobom ako prebieha detekcia celých objektov. Rozdiel spočíva v tom, že do selection bufferu nie sú kreslené celé objekty, ale len štvorčeky zvýrazňujúce ich body. Na zásobník sú teda ukladané len mená týchto bodov, sú to vlastne mená objektov, ktorým body patria, doplnené o identifikátor bodu (stred, vrchol, ...). Ak je bod následne „potiahnutý“, tak je volaná metóda **dragNdrop**. Na základe poskytnutého mena bodu a pozície kurzoru sú prepočítané súradnice tohoto bodu patriaceho príslušnému objektu. Ak je týmto bodom stred, tak je objekt posunutý celý, inak sa mení geometria objektu (polomer, veľkosť strán, ...).

3.4.3 Uchopovací mód

Významné body objektov, okrem toho, že sa dá meniť ich pozícia, plnia ďalšiu dôležitú funkciu – je možné ich k sebe uchopovať. Ak zadáme napr. počiatočný bod úsečky dostatočne blízko k vrcholu obdĺžnika, tak úsečka bude mať začiatok v tomto vrchole. Toto bol konkrétny príklad, akýkoľvek zadaný bod je možné takýmto spôsobom uchopiť k iným významným bodom. Typy bodov, ktorým možno uchopovať:

- koncové body – vrcholy obdĺžnika, krajné body priamky, krajné body krivky
- kvadrantové body kružnice a elipsy
- stredy – jednotlivých objektov, strán obdĺžnika
- priesečníky

Návrh zahŕňa tiež 2 typy uchopenia, špecifické pre úsečku:

- päta kolmice – úsečka je zostrojená ako kolmica na inú úsečku alebo obdĺžnik
- dotyčnica – úsečka je zostrojená ako dotyčnica ku kružnici alebo elipse

Hľadanie vhodných bodov prebieha pri každom pohybe kurzoru, čo môže byť výpočetne veľmi náročné. Preto je vhodné nehľadať body na všetkých objektoch, ale len na tých, ktoré sú najbližšie ku kurzoru. Podľa nastavenia uchopovacieho módu sú zvýraznené vyhovujúce body na blízkych objektoch.

Alternatívnym riešením zjednodušenia hľadania bodov, by mohlo byť rozdelenie plátna na rovnako veľké časti. Každá z týchto častí by si uchovávala zoznam objektov, ktoré sa v nej nachádzajú alebo do nej zasahujú. Pri aktívnom uchopovacom móde by bola vždy prehľadávaná len tá časť, v ktorej by sa nachádzal kurzor. Toto riešenie by však malo význam len pri rozdelení plátna na dostatočne veľký počet častí.

Programová realizácia detekcie bodov a uchopenia

Ako sme povedali, vyhovujúce body sú hľadané len na najbližších objektoch. Tieto objekty sú detekované rovnakým spôsobom ako u výberového módu, pomocou selection bufferu v okolí kurzoru $10 * 10$ pixelov. Detekciu vykonáva metóda `nearestPoint` triedy `Attachment`. Najprv nájde blízke objekty, a potom na základe nastavenia módu pridáva do zoznamu ich vyhovujúce body. Ak je aktívne uchopovanie úsečky ako kolmice, tak body, ktoré môžu byť jej pätou, vypočíta funkcia `verticalLine` a pridá ich do zoznamu. Body dotyku ku kružnici sú počítané funkciou `tangentLineCircle` a k elipse funkciou `tangentLineEllipse`.

V prípade, že sa bude uchopovať k priesečníkom, tak pre každú dvojicu blízkych objektov je volaná špecifická funkcia, podľa ich typu. Celkovo sa jedná o 6 funkcií:

- `intersectionLineLine` – priesečníky 2 úsečiek. Keďže obdĺžniky a krivky sú množinami úsečiek, je táto funkcia je požívaná aj pri hľadaní priesečníkov úsečky a obdĺžnika, úsečky a krivky, obdĺžnika a krivky, 2 obdĺžnikov a 2 kriviek.
- `intersectionLineCircle` – priesečníky úsečky a kružnice, funkcia je používaná aj na vypočítanie priesečníkov obdĺžnika a kružnice a krivky a kružnice.

- `intersectionLineEllipse` – priesečníky úsečky a elipsy, funkcia slúži tiež na získanie priesečníkov obdĺžnika a elipsy a krivky a elipsy.
- `intersectionCircleCircle` – priesečníky 2 kružníc.
- `intersectionCircleEllipse` – priesečníky kružnice a elipsy.
- `intersectionEllipseEllipse` – priesečníky 2 elíps.

Výpočty vychádzajú z analytických vyjadrení geometrických útvarov, čerpaných zo stránok [14], [2] a [15]. Obdĺžnik a krivka nemajú v systéme analytické vyjadrenie, sú však množinami úsečiek. Priesečníky s nimi sú teda počítané ako priesečníky každej úsečky tvoriacej obdĺžnik, resp. krivku s daným objektom. Získané priesečníky sú pridané do zoznamu vyhovujúcich bodov. Keď sú prehľadané všetky blízke objekty, tak zo zoznamu je vybraný bod najbližší kurzoru. Nestačí však, aby bol najbližšie, musí byť aj dostatočne blízko ku kurzoru. Ak bod spĺňa aj túto podmienku, tak ho funkcia vráti, ako bod pre uchopenie, inak vráti pôvodnú pozíciu kurzoru.

3.4.4 Transformácie vybraných objektov

Uviedli sme si, že vyznačené objekty možno do istej miery transformovať zmenou pozície ich významných bodov. Tento spôsob nám však neumožňuje objekty transformovať s požadovanou presnosťou a rotovať, preto je v aplikácii navrhnuté zadávanie transformácií pomocou užívateľského rozhrania. Po spracovaní a kontrole sú hodnoty poslané ako parametre do transformačných funkcií, ktoré sú implementované triedou `Canvas`.

Posun s možnosťou kopírovania

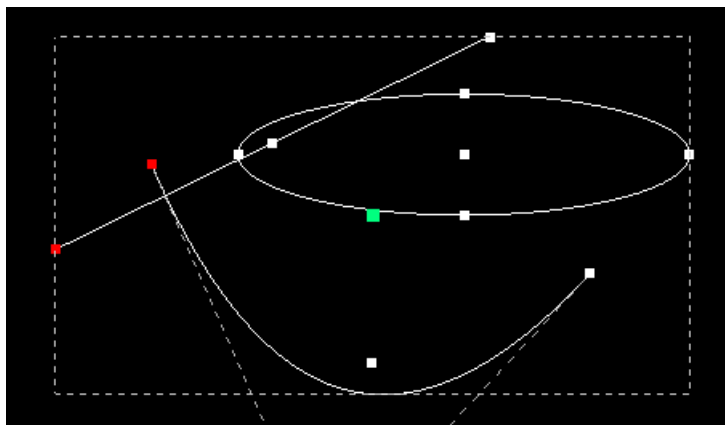
Posun prebieha na základe zadaných súradníc vektoru. Majme určený vektor posunutia $\vec{v}[5, -4]$. Jeho súradnice nám hovoria, že objekt bude posunutý o 5 jednotiek dĺžky na osi x a o -4 jednotiek dĺžky na osi y . Návrh zahŕňa taktiež variantu kopírovania, kedy sa postupuje rovnako ako pri posúvaní, ale pôvodný objekt zostane na svojom mieste. Na pozícii, kam by bol posunutý, bude vytvorená jeho kópia. Tento postup môžeme aplikovať aj na kopírovanie do inej hladiny. Ak vyberieme objekt z inej hladiny ako je práve aktívna a skopírujeme ho, tak kópia bude patriť do práve aktívnej hladiny (aj neviditeľnej), čomu bude zodpovedať aj jeho farba.

Posun vykonáva funkcia `translateSelected`, ktorá ako parametre dostane súradnice vektoru posunutia a príznak indikujúci, či má funkcia posúvať alebo kopírovať. Pri posune budú súradniciam každého významného bodu objektu pripočítané súradnice vektoru posunutia a po prekreslení plátna bude objekt posunutý. Ak budeme kopírovať, tak je potrebné vytvoriť nový objekt s geometriou kopírovanej predlohy a posunúť ho na základe vektoru posunutia.

Rotácia o zadaný uhol v stupňoch

Uhol rotácie je zadávaný buď z klávesnice, alebo pomocou rotačného tlačítka, ktoré bude bližšie popísané v sekcii 3.5 venovanej návrhu užívateľského rozhrania. Návrh umožňuje zadávať uhol len v jednotkách stupňov a celých číslach. V potenciálnych rozšíreniach aplikácie by sa preto mohla objaviť varianta zadávania uhlu v radiánoch a desatinných číslach. V prípade, že je vybraný len 1 objekt, tak je rotovaný okolo svojho stredu, pokiaľ nezadáme iný bod. Ak vyberieme viac objektov, tak z ich krajných bodov je vypočítaný ich

potenciálny stred, ako znázorňuje obrázok 3.4. Čiarkovaný obdĺžnik je len pre ilustráciu krajných bodov vyznačených objektov. Svetlozelený bod znázorňuje stred rotácie je stredom čiarkovaného obdĺžnika. Aplikácia užívateľovi dovoľuje zmeniť stred rotácie pomocou kombinácie M + ľavé tlačítko myši. Objekty si neuchovávajú hodnotu uhlu rotácie (okrem obdĺžnika a elipsy), a preto keď vyberieme objekt, rotujeme ho napr. o 30 stupňov, tak pri jeho opätovnom vybratí sa v rozhraní nenastaví 30, ale 0 stupňov.



Obrázok 3.4: Svetlozelený bod znázorňuje stred rotácie viacerých vybraných objektov

Funkcia `rotateSelected`, ktorá má rotáciu na starosti, má ako parametre nový a predchádzajúci uhol rotácie. Ich rozdielom vypočítame, o aký uhol budeme rotovať všetky významné body vybraných objektov okolo stredy rotácie. Body sú v princípe rotované po kružnici, ktorej stredom je stred rotácie a polomer sa rovná vzdialenosti stredy a transformovaného bodu. Každému bodu teda prislúcha vlastná kružnica, po ktorej bude rotovaný o zadaný uhol.

Zmena veľkosti objektu zadávaná v percentách

Aplikácia umožňuje zmenšovať alebo zväčšovať všetky objekty naraz, no zároveň sa pritom mení aj viditeľný rozsah súradnicovej sústavy. Návrh však počíta s tým, že užívateľ bude chcieť meniť len veľkosť vybraných objektov bez toho, aby menil rozsah sústavy. Pri vybratí objektu je hodnota nastavená na 100%, pričom jej rozsah je 1–999%. U objektov, ktoré tvoria určitú plochu (obdĺžnik, kružnica, elipsa) nie je podľa zadaného pomeru zmenená táto plocha, ale ich rozmery. Ak chceme zmenšiť každý z týchto objektov na 70%, tak u obdĺžnika dôjde skráteniu všetkých jeho strán, u kružnice polomeru a u elipsy jej poloosí na 70% pôvodnej dĺžky. V prípade, že vyberieme viac objektov a budeme meniť ich veľkosť, tak bude každý menený nezávisle na tom, koľko objektov je vybraných a aká je ich pozícia.

Zmena je realizovaná pomocou funkcie `scaleSelected`. Spôsob, akým funkcia pracuje, závisí od typu objektu, ktorý bude meniť.

- **Úsečka** – mení sa poloha oboch krajných bodov. Pre bod sú na osi x a y vypočítané vzdialenosti od stredy, následne sú tieto vzdialenosti zmenené na základe zadaného parametru. Aplikovaním tohoto postupu na oba krajné body dosiahneme zmenu dĺžky úsečky podľa zadaného parametru.
- **Obdĺžnik** – rovnaký postup ako pri úsečke aplikujeme na uhlopriečky, čím sa každá strana obdĺžnika zmení podľa zadanej hodnoty, nakoniec sú prepočítané stredy strán

- **Kružnica** – zmenená je veľkosť polomeru.
- **Elipsa** – keďže aj elipsa si uchováva informácie o vrchoch obdĺžnika, ktorému je vpísaná, použijeme rovnaký postup ako u obdĺžnika na zmenu pozície vrcholov a prepočítame kvadrantové body a veľkosti poloôs.
- **Krivka** – zmenené sú vzdialenosti kontrolných bodov od stredu krivky.

3.4.5 Ortogonálny mód

Kreslenie úsečiek rovnobežných s osami súradnicovej sústavy je pri vytváraní technických výkresov pomerne časté, a preto návrh zahŕňa aj túto variantu ich vykresľovania. Aktívny môže byť vždy len 1 z 2 „podmódov“:

Horizontálny

Úsečka je kreslená rovnobežne s osou x . Keď je pri obsluhu udalostí myši na plátne aktívny tento mód, tak si uložíme počiatočný bod. Súradnicu y nastavujeme koncovému bodu pri dynamickom kreslení a vytváraní objektu vždy rovnakú ako má počiatočný bod. Súradnicu x bude mať koncový bod rovnakú ako kurzor.

Vertikálny

Úsečka je kreslená rovnobežne s osou y . Keď je pri obsluhu udalostí myši na plátne aktívny tento mód, tak si uložíme počiatočný bod. Súradnicu x nastavujeme koncovému bodu pri dynamickom kreslení a vytváraní objektu vždy rovnakú ako má počiatočný bod. Súradnicu y bude mať koncový bod rovnakú ako kurzor.

3.4.6 Práca so súborom

Navrhovaná aplikácia dovoľuje užívateľovi uložiť rozrobenú prácu do súboru, resp. ju zo súboru načítať. Do súboru sú ukladané dáta – informácie o:

- plátne – názov, cesta k súboru, v ktorom je plátno uložené
- hladinách – názov, farba, viditeľnosť
- objektoch – body, dĺžky a všetky údaje potrebné pre správne vykreslenie

Základom práce so súbormi je trieda `Filework` a jej metódy `saveFile` a `loadFile`. Pred samotnou (de)serializáciou je v týchto metódach potrebné otvoriť (vytvoriť) súbor a priradiť mu dátový tok, ktorý je objektom triedy `QDataStream`. Potom je zavolaná funkcia na (de)serializáciu plátna, ktorá (de)serializuje informácie o plátne a volá funkciu na (de)serializáciu každej hladiny. Tá (de)serializuje svoje informácie a postupne všetky jej objekty. Operátor `<<` nám umožňuje serializovať dáta do toku, vždy je nutné uviesť aj ich dátový typ, napr. `stream << (quint64)ID`. Deserializáciu z toku realizujeme pomocou operátoru `>>`, pričom už nie je nutné uvádzať dátový typ, napr. `stream >> ID`. Dáta je nutné čítať z toku v rovnakom poradí ako doňho boli ukladané, inak by mohlo dôjsť k načítaniu nesprávnych dát.

Aplikácia musí vedieť, či otvorené plátno bolo zmenené alebo nie, aby pri jeho zatváraní nedošlo k strate vykonaných zmien. Preto si každé plátno uchováva príznak `saved`, určujúci

či plátno obsahuje neuložené zmeny, alebo nie. Akcie, ktoré spôsobia, že plátno bude obsahovať neuložené zmeny sú:

- Vytvorenie alebo zmazanie objektu.
- Vytvorenie alebo zmazanie hladiny, zmena viditeľnosti alebo farby hladiny
- Transformácie vybraných objektov

Ukladať dáta len v serializovanej podobe navrhnutej výhradne pre túto aplikáciu je nepraktické. V rozšíreniach aplikácie by sa určite mal objaviť export do rôznych formátov, napr. XML, PDF, SVG, ako súbor vektorovej grafiky alebo DXF pre zaistenie prenositeľnosti medzi rôznymi CAD systémami.

3.5 Grafické užívateľské rozhranie

CAD systém, ako program na kreslenie potrebuje grafické užívateľské rozhranie, ktoré je dostatočne prehľadné a jednoducho ovládateľné. Systém je navrhnutý ako oknová aplikácia. Trieda `MainWindow` reprezentujúca hlavné okno je potomkom triedy `QMainWindow`, ktorej použitie umožňuje modul `QtGui` určený pre tvorbu aplikácie s užívateľským rozhraním. Tá nám poskytuje možnosti budovať vlastné rozhranie, ktorého jednotlivé časti si v tejto sekcii popíšeme.

Rozhranie je zložené z 8 boxov, ktoré sú objektmi triedy `QGroupBox`. Všetky textové políčka užívateľského rozhrania sú objekty triedy `QLineEdit`. Je možné do nich vkladať príslušné hodnoty (na nevyhovujúce je užívateľ upozornený) a potvrdzovať ich klávesou *Enter*.

3.5.1 Scéna, výber entity a štýly čiar

Najdôležitejšou časťou rozhrania je scéna, kam užívateľ môže kresliť a je na nej zobrazené práve aktívne plátno. Scéna je navrhnutá tak, aby s ňou bola umožnená užívateľská interakcia, čiže užívateľ môže pridávať alebo odoberať objekty a transformovať ich, meniť priblíženie a posunutie plátna a presne uchopovať body.

Predtým, ako bude užívateľ kresliť, musí si vybrať entitu. Na výber slúži 7 tlačítiek s ikonou vykresľovaného objektu alebo výber v hlavnom menu aplikácie (*Edit* → *Shape*). Aktívne môže byť vždy vykresľovanie len 1 objektu. Tlačítka sú zoskupené v boxe *Shapes*.

Box *Line styles* obsahuje 4 tlačítka na zmenu štýlu čiar vykresľovaných objektov. Na kliknutie reaguje príslušný slot zmenou štýlu pre všetky následne nakreslené objekty. Boxy sú znázornené na obrázku 3.5. Štýl čiar možno meniť aj v hlavnom menu (*Edit* → *Line style*)

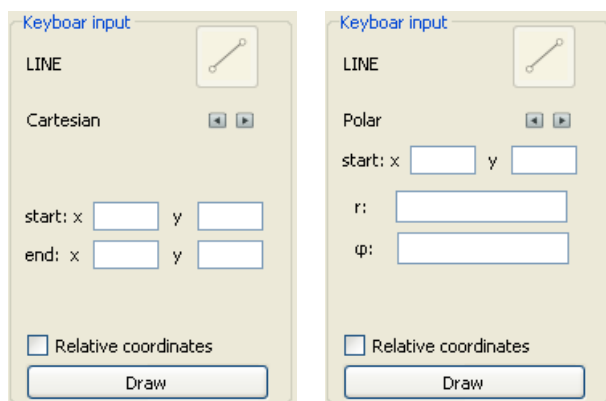


Obrázok 3.5: Boxy pre výber entít a štýlov čiar

3.5.2 Zadávanie parametrov klávesnicou

S výberom vykresľovanej entity sa mení box *Keyboard input* nachádzajúci sa pod tlačítkami určenými pre tento výber. Jeho obsah tvorí objekt triedy `QStackedWidget`, zásobník panelov, medzi ktorými sa dá pomocou šípiek prepínať. Tento zásobník obsahuje 2 panely, no prepínať medzi nimi sa dá len v prípade, že je aktívne kreslenie úsečky. Prvý panel (obr. 3.6(a)) je prístupný pre všetky entity a umožňuje užívateľovi zadávať presné súradnice a dĺžky (polomer, poloosi) pri kreslení. Druhý panel (obr. 3.6(b)) obsahuje políčka na zadávanie úsečky pomocou polárnych súradníc, kedy je pre vykreslenie nutné zadať súradnice jej počiatočného bodu, dĺžku a odchýlku (uhol v stupňoch) od osi x . Ako budúce rozšírenie, resp. by sa jednalo prevažne o modifikáciu, možno uvažovať nad zovšeobecnením zadávania parametrov. Namiesto textových políček by existoval jeden príkazový riadok, ktorý by prijímal makrá v skriptovacom jazyku alebo jednotlivé parametre po jednom. Tým by bolo umožnené aj zadávanie riadiacich bodov krivky z klávesnice.

V paneli na kreslenie z klávesnice môže užívateľ nastaviť režim zadávania relatívnych súradníc. V praxi to znamená, že aplikácia si zapamätá posledný zadaný bod (u kružnice a elipsy to bude vždy stred) a ten bude pri nasledovnom zadávaní parametrov z klávesnice považovaný za relatívny počiatok súradnicovej sústavy.



(a) Zadávanie kartézskych súradníc (b) Zadávanie polárnych súradníc

Obrázok 3.6: Klávesnicové zadávanie

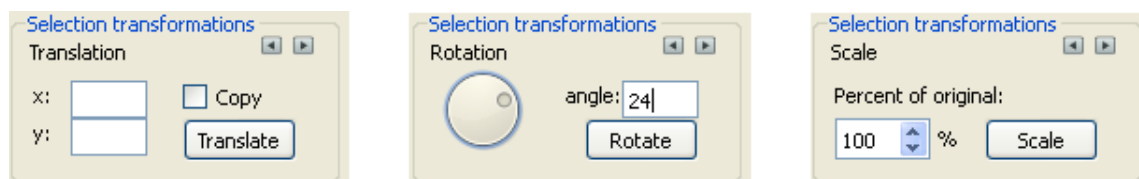
3.5.3 Transformácie výberu

Box *Selection transformations* slúži na vykonávanie transformácií vybraných objektov. Jeho obsahom je tiež objekt triedy `QStackedWidget`, v ktorom sa nachádzajú 3 alternatívne panely pre:

- posun – 2 textové políčka pre zadanie súradníc vektoru posunu, zaškrŕavacie políčko pre voľbu kopírovania
- rotáciu – 1 textové políčko pre zadanie uhlu v stupňoch, interaktívne otočné tlačítko
- zmenu veľkosti – 1 textové políčko s tlačítkami pre zmenu hodnoty políčka o 1%

Každý panel obsahuje tlačítko na potvrdenie zadaných údajov, na jeho stlačenie reaguje pre každú transformáciu špecifický slot. Spracované a skontrolované hodnoty sú predané plátnu na vykonanie príslušnej transformácie. Panely znázorňuje obrázok 3.7.

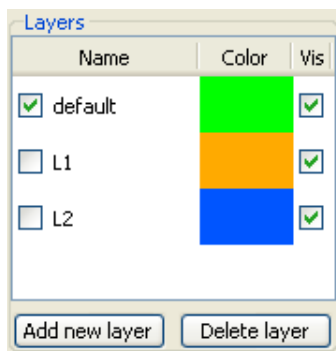
Otočné tlačítko v paneli pre rotáciu je objekt triedy `QDial` a možno ho obsluhovať viacerými spôsobmi. Ľavým tlačítkom myši rotujeme objekt rýchlo, no nie príliš presne. Pre presnú rotáciu použijeme koliečko myši, resp. klávesy *PageUp* a *PageDown*. Pri 1 pootočení koliečka smerom od seba, resp. stlačení klávesy *PageUp* sa zvýši stupeň rotácie o 1. Točenie smerom k sebe, resp. klávesa *PageDown* uhol o 1 stupeň zníži. Na každé pootočenie reaguje slot `on_dial_valueChanged`, ktorý údaje spracuje a posielá ich ďalej plátnu na rotáciu.



Obrázok 3.7: Boxy pre vykonávanie transformácií

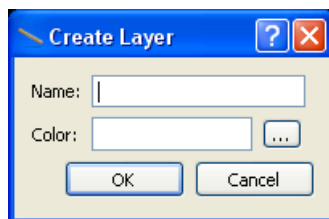
3.5.4 Tabuľka hladín

Tabuľka je objektom triedy `QTableWidget` a nachádza sa v boxe *Layers* (obr. 3.8). Skladá sa z 3 stĺpcov a vopred neurčeného počtu riadkov. V 1. stĺpci je názov hladiny a zaškrŕtávacie políčko pre jej aktiváciu. Názov hladiny nie je možné meniť. Bunka 2. stĺpca je prakticky prázdna, mení sa len jej pozadie na základe vybranej farby. Farbu možno zmeniť dvojkliknutím na bunku v 2. stĺpci, čím sa nám otvorí dialóg pre výber farby. Zaškrŕtávacie políčko v 3. stĺpci slúži na zmenu viditeľnosti príslušnej hladiny. Obsluhu udalostí nad tabuľkou majú na starosti sloty implementované triedou `MainWindow`.



Obrázok 3.8: Tabuľka hladín

Pod tabuľkou sú umiestnené 2 tlačítka, jedno na pridanie a druhé na zmazanie hladiny. Po kliknutí na tlačítko pre pridanie sa zobrazí malé dialógové okno znázornené na obrázku 3.9. Pridávať a odstraňovať hladiny je možné aj v hlavnom menu aplikácie (*Edit* → *Layers*).



Obrázok 3.9: Dialóg pre vytvorenie novej hladiny

Do 1. textového políčka zadáme názov hladiny a tlačítkom vedľa 2. políčka otvoríme dialóg pre voľbu farby. Pozadie 2. políčka je vyfarbené zvolenou farbou. Tlačítkom *Ok* vytvoríme hladinu, *Cancel* naopak vytváranie zruší. Nie je možné vytvoriť hladinu bez názvu alebo s názvom, ktorý má inú, už existujúcu, hladinu.

3.5.5 Uchopovací mód

Box *Point attachment* (obr. 3.10) pre uchopovací mód pozostáva len zo zaškrŕavacích políčk. To najdôležitejšie je vedľa nadpisu panelu a slúži na aktiváciu, resp. deaktiváciu módu. Po jeho zaškrŕnutí aplikácia užívateľovi sprístupní ďalších 5 políčk pre zadanie typu bodov, ku ktorým bude chcieť uchopovať. Políčka *vertical* a *tangent* sú prístupné len vtedy, ak je aktívne kreslenie úsečiek. Ostatné políčka sú prístupné pre všetky entity, ak je aktívny mód. Zaškrŕvaním a odškrŕvaním políčk príslušné sloty nastavujú pole príznakov pre metódu `Attachment::nearestPoint`. V hlavnom menu je tiež obsluha uchopovacieho módu (*Edit* → *Point attachment*).

3.5.6 Ortogonálny mód

Rovnako, ako u uchopovacieho, možno aj ortogonálny mód aktivovať a deaktivovať zaškrŕvacím políčkom alebo obsluhovať v hlavnom menu (*Edit* → *Orthogonal mode*). Pokiaľ je mód aktívny, užívateľ môže vyberať medzi horizontálnou a vertikálnou alternatívou. Neaktívny mód spôsobí, že výber medzi alternatívami nebude možný. Obsah boxu *Ortho mode* (obr. 3.10) tvoria tzv. „rádiové tlačítka“, z ktorých je vždy aktívne práve jedno.

3.5.7 Priblíženie plátna

Zoom canvas (obr. 3.10) je box umožňujúci obsluhu priblíženia plátna. Obsahuje editovateľné textové políčko s číslom znázorňujúcim aktuálne priblíženie a 3 tlačítka. 2 menia priblíženie po krokoch 5% a tretie nastaví priblíženie na pôvodných 100%. Obsluhu priblíženia je možné vykonávať aj cez hlavné menu (*Edit* → *Zoom*).



Obrázok 3.10: Boxy pre uchopovací mód, ortogonálny mód a priblíženie plátna

3.6 Implementačné prostriedky

Pri voľbe programovacieho jazyka, implementačných prostriedkov a nástrojov bol braný ohľad na predošlé skúsenosti a prenositeľnosť aplikácie. Ako programovací jazyk je použitý objektovo orientovaný jazyk C++. Pre správu verzií a zálohovanie zdrojových kódov bol použitý nástroj TortoiseSVN [13].

3.6.1 OpenGL

Voľba knižnice bola dôležitá z hľadiska prenositeľnosti aplikácie, preto bolo zvolené rozhranie OpenGL – Open Graphics Library. Popis vychádza z knihy [12].

OpenGL je softwarové rozhranie pre grafický hardware. Pozostáva z viac než 700 rôznych príkazov, ktoré používame na špecifikáciu objektov a operácií pri vytváraní interaktívnych aplikácií. Toto rozhranie je hardwarovo nezávislé, aby mohlo byť použité na implementáciu aplikácií na rôznych hardwarových platformách. Z toho vyplýva, že OpenGL neobsahuje žiadne príkazy na správu okien a obsluhu užívateľských vstupov, takže je nutné pracovať s dostupnými hardwarovo podporovanými systémami pre správu okien. OpenGL taktiež neposkytuje príkazy na popis 3D modelov, je nutné ich popísať pomocou geometrických primitív – body, čiary a polygóny. Pozostáva však z mnohých stavov a stavových premenných – prakticky je to stavový automat. Objekty môžu byť vykreslené s osvetlením, textúrami, hmlou a inými stavmi ovplyvňujúcimi ich vzhľad. Väčšina stavov je na začiatku neaktívna a za cenu zníženia výkonu môžu byť postupne aktivované. Napríklad zapnutie mapovania textúr takmer určite spomalí proces vykresľovania primitív, no na druhej strane zlepší ich vizuálnu kvalitu a realnosť.

V navrhovanej aplikácii je OpenGL použité na inicializáciu scény, nastavenie jej projekcie, kreslenie a výber objektov scény. Každý objekt je kreslený pomocou OpenGL buď ako jedna čiara, alebo konečná množina čiar. Kľúčovú úlohu zohráva OpenGL pri implementácii výberového 3.4.2 a uchopovacieho módu 3.4.3.

3.6.2 Qt

Pre tvorbu užívateľského rozhrania bola zvolená rozsiahla knižnica Qt. Tá poskytuje možnosť tvoriť multiplatformné aplikácie spustiteľné na počítačoch, mobiloch aj vstavaných operačných systémoch bez nutnosti prepisovať zdrojový kód [9]. Nezanedbateľnými faktormi pri voľbe boli aj možnosť použitia Qt s jazykom C++, predošlé skúsenosti s knižnicou Qt a rozsiahla, prehľadná dokumentácia, ktorá je voľne dostupná na internete [10]. Pre vývoj aplikácie bolo použité integrované vývojové prostredie Qt Creator 1.3.1 a Qt SDK s knižnicami verzie 4.6.2. Prostredie ponúka možnosť využiť *WYSIWYG* (*what you see is what you get*) nástroj Qt Designer, ktorý slúži na vytváranie grafického užívateľského rozhrania aplikácie.

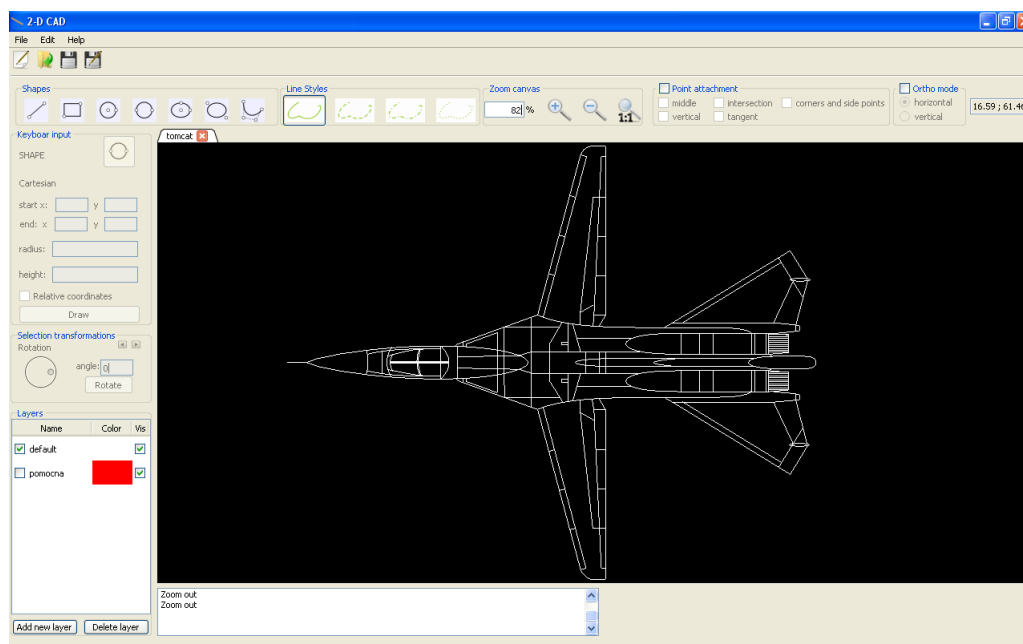
Kapitola 4

Výsledky

Jednou z priorit tejto práce bolo vysvetliť čo je to CAD systém a popísať požiadavky, aké funkcie by mal poskytovať. Týmto sa zaoberá kapitola 2. Hlavnou úlohou však bolo navrhnúť a naimplementovať CAD systém pre 2D kreslenie podľa stanovených cieľov 3.1 tak, aby bol jednoducho použiteľný a rozšíriteľný o nové entity a funkcie. V tejto kapitole si zhrnieme aké sú vlastnosti a funkcie vytvoreného systému.

4.1 Systém ako aplikácia

CAD systém pre 2D kreslenie je implementovaný ako oknová aplikácia s grafickým užívateľským rozhraním.



Obrázok 4.1: Ukážka aplikácie

Rozhranie bolo navrhnuté a vytvorené tak, aby užívateľ mohol jednoducho vytvárať nové objekty, organizovať ich do hladín a spravovať plátna. Všetky funkcie sú dostupné

z rozhrania a sú rozložené tak, aby boli jednoducho ovládateľné. Určité akcie majú alternatívy aj v podobe klávesových skratiek. Jedná sa najmä o prácu so súborom, ktorej ikony sú umiestnené pod hlavným menu okna. Tieto akcie užívateľovi umožňujú vytvárať nové plátna ($Ctrl + N$), ukladať ($Ctrl + S$) a otvárať plátna ($Ctrl + O$). V hlavnom menu okna sa v časti *File* nachádzajú práve spomenuté akcie a navyše akcia ukončenia aplikácie ($Ctrl + Q$). V časti *Edit* sú akcie spojené s určovaním objektov, uchopovacím a ortogonálnym módom, štýlom čiar, správou hladín a priblížením plátna. Sekcia *Help* obsahuje len 1 akciu, ktorá otvorí v predvolenom internetovom prehliadači stručný manuál k aplikácii vo formáte PDF. Na obrázku 4.1 môžeme vidieť konečný vzhľad aplikácie s nákrešom, ktorý bol pomocou nej vytvorený.

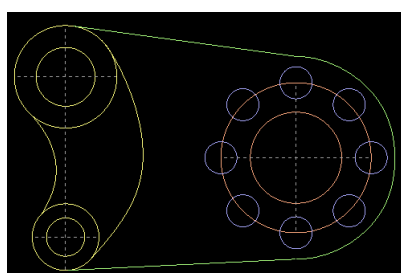
4.2 Entity a dátové kontajnery

Implementované entity (spôsob ich zadávania je uvedený v sekcii 3.2.1):

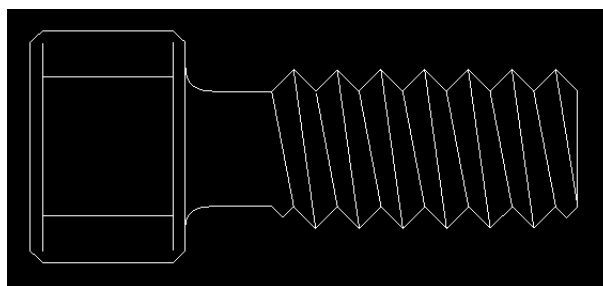
- úsečka
- obdĺžnik
- kružnica
- elipsa
- krivka

Hladiny

Aplikácia disponuje rozdelením plátna do hladín, čím užívateľovi umožňuje organizovať objekty do logických celkov. Hladiny majú unikátne názvy, možno ich pridávať a mazať, pričom hladina *default* je nezmazateľná. Limit existujúcich hladín je 999. Objekty 1 hladiny majú rovnakú farbu, ktorú užívateľ môže ľubovoľne meniť. Každá hladina má nastaviteľnú viditeľnosť, jej skrytím budú skryté všetky jej objekty. Objekty sú kreslené do aktívnej hladiny, nemôžu byť kreslené samostatne, bez príslušnosti k hladine. Aktívna je vždy práve 1 hladina, implicitne je to *default*. Názorná ukážka práce s hladinami je na obrázku 4.2(a). Jednotlivé hladiny sú farebne rozlíšené.



(a) Ukážka organizácie do hladín



(b) Ukážka nákresu konkrétneho predmetu – skrutky

Obrázok 4.2: Ukážky nakreslené vytvorenou aplikáciou

Plátna

Rovnako ako hladiny zoskupujú objekty, tak plátna zoskupujú hladiny. Užívateľ môže vytvárať nové plátna, zatvárať existujúce, ukladať plátno do súboru v binárnej podobe a načítať ho zo súboru. Aplikácia je schopná mať otvorených viac plátien, užívateľ má možnosť prepínať medzi nimi, aktívne môže byť len jedno. Každé plátno má pri vytvorení implicitný názov *untitled*, po uložení do súboru sa názov prepíše na názov súboru. Maximálny limit práve otvorených plátien je 20.

4.3 Transformácie plátna

Transformácie posunu a priblíženia plátna sú vykonávané pomocou myši a klávesnice. Pre posun užívateľ podrží klávesu *Ctrl* a ľavým tlačítkom myši „potiahne“ plátno. Zmenu priblíženia umožňuje koliesko myši, keď je kurzor nad plátnom, klávesové skratky *Ctrl* + + a *Ctrl* + - alebo tlačítkami užívateľského rozhrania. Krok je 5%, rozsah je 0–999%. Pôvodné priblíženie 100% nastavíme príslušným tlačítkom v užívateľskom rozhraní alebo klávesovou skratkou *Ctrl* + 0.

4.4 Výberový mód

Aplikácia obsahuje mód, v ktorom užívateľ vyberá objekty plátna pomocou myši klikaním na ne alebo obdĺžnikom. Obdĺžnikom sú vybrané všetky objekty, ktoré doň aspoň čiastočne zasahujú. Objekty sa dajú do výberu pridávať alebo z neho odoberať, keď je počas výberu držaná klávesa *Shift*. Klávesou *del* je možné vybrané objekty zmazať. Mód taktiež zahŕňa obsluhu výberu a posunutia významných bodov objektov alebo objektov samotných.

4.5 Transformácie výberu

Aplikácia dovoľuje vybrané objekty transformovať 3 spôsobmi:

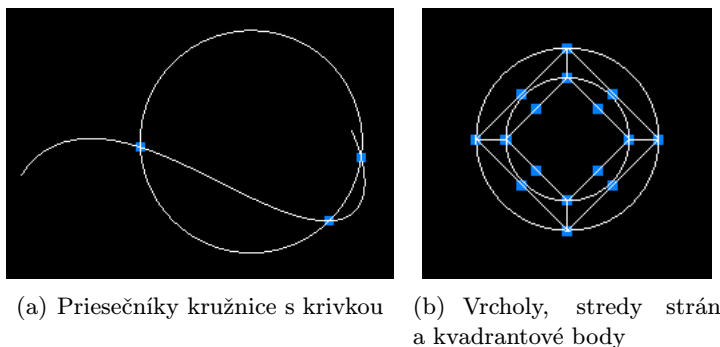
- posun s možnosťou kopírovania, kopírovania aj do inej hladiny
- rotácia okolo stredu vybraných objektov alebo stredu zadaného užívateľom pomocou kombinácie klávesy *m* a ľavého tlačítka myši, uhol rotácie je v stupňoch
- zmena veľkosti udávaná v percentách

4.6 Uchopovací mód

Pre zvýšenie presnosti systému bol do aplikácie implementovaný aj uchopovací mód, ktorý užívateľovi dovoľuje presne uchopovať body pri kreslení objektov alebo posúvaní ich významných bodov. Body, ku ktorým sa dá uchopovať:

- koncové body – vrcholy obdĺžnika, krajné body priamky, krajné body krivky
- kvadrantové body kružnice a elipsy
- stredy – jednotlivých objektov, strán obdĺžnika
- priesečníky

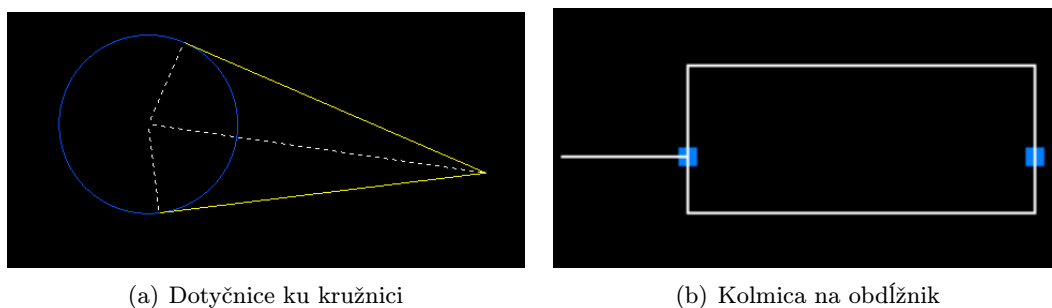
Na obrázku 4.3(a) sú zvýraznené priesečníky krivky a kružnice a na obrázku 4.3(b) sú zvýraznené vrcholy a stredy strán obdĺžnikov (v tomto prípade štvorcov) a kvadrantové body kružníc. Body sú zvýraznené tak, ako ich aplikácia zvýrazňuje pri práci.



Obrázok 4.3: Zvýraznenie bodov na uchopenie

2 typy uchopenia, špecifické pre úsečku:

- päta kolmice – úsečka je zostrojená ako kolmica na inú úsečku alebo obdĺžnik, obr. 4.4(b)
- dotyčnica – úsečka je zostrojená ako dotyčnica ku kružnici alebo elipse, obr. 4.4(a)



Obrázok 4.4: Dotyčnice a kolmica

4.7 Ukladanie a načítanie práce

Rozpracovaný náčrt je možné uložiť do súboru a pre neskoršie pokračovanie v práci ho zo súboru opätovne načítať. Dáta sú ukladané v binárnej podobe navrhutej pre túto konkrétnu aplikáciu. Ukladanie prebieha postupne od informácii o plátne, cez informácie o hladinách, až po samotné grafické entity.

Kapitola 5

Záver

Výsledkom práce je vlastný návrh a implementácia CAD systému, ako multiplatformnej, jednoducho rozširiteľnej a použiteľnej aplikácie.

Okrem samotnej aplikácie práca zahŕňa tiež teoretický rozbor danej problematiky. Práca čitateľa oboznamuje s významom a výhodami použitia počítačov v priemyselnej výrobe, popisuje metódy a postupy, ktoré sú v dnešnej dobe z veľkej časti podporované počítačmi. Najväčší dôraz kladie na CAD systémy, sú uvedené ich rozdelenie z pohľadu rozsahu a základné požiadavky na ich funkcionality.

V úvode návrhovej fázy som stanovil ciele vychádzajúc z informácií a požiadaviek uvedených v teoretickom rozbere. Implementačná fáza, kedy bol použitý programovací jazyk C++ v kombinácii s Qt knižnicami pre tvorbu grafického užívateľského rozhrania a OpenGL pre grafický výstup, bola realizovaná tak, aby som čo najpresnejšie splnil stanovené ciele. Voľba implementačných nástrojov sa ukázala ako vhodná pre tvorbu tejto aplikácie. Výsledkom implementácie je jednoduchý CAD systém, poskytujúci užívateľovi možnosť kresliť základné entity, organizovať ich do hladín a vykonávať základné transformácie objektov. Systém taktiež podporuje rôzne alternatívy vytvárania objektov a módy pre uľahčenie a urýchlenie ich tvorby.

Celkový prínos práce spočíva, viac než v použiteľnosti aplikácie v praxi, v oboznámení sa s problematikou CAD systémov a získaní nových skúseností s knižnicami Qt a OpenGL.

5.1 Možnosti budúcej práce

Systém poskytuje len zlomok funkcionality profesionálnych systémov, a preto je vhodné uvažovať o budúcich rozšíreniach. V prvom rade by sa týkali entít, konkrétne ide napr. možnosť kresliť lomenú čiaru, kružnicu zadáním 3 bodov, vytvárať NURBS krivky alebo kružnicové oblúky. Kreslenie by tiež mohlo byť realizovateľné s rôznymi jednotkami dĺžky a klávesnicové zadávanie parametrov by bolo dobré zovšeobecniť na zadávanie makier alebo príkazov namiesto zadávania každého parametru do zvláštného políčka. Transformácie by mali byť doplnené o zrkadlenie a skosenie a rotácia o zadávanie uhlov v radiánoch a desatinných číslach. Aplikácia ponúka prácu so súborom, no len s binárnym, preto by v budúcnosti mohla byť doplnená o rôzne formáty súborov pre export a import (DXF, XML, ...). Programy slúžiace na systematickú tvorbu niečoho konkrétneho, disponujú funkciami krok späť a krok vpred. Vytvorený systém tieto funkcie nepodporuje, no v budúcnosti by určite mal.

Literatúra

- [1] BLIŠŤAN, P.: *Úvod do počítačovej grafiky a CAD systémov*. Košice: Editačné stredisko AMS, prvé vydání, 2004, ISBN 80-8073-249-3, [online], [cit. 2011-05-07]. URL <http://www.fberg.tuke.sk/blistan/Pocitacova%20grafika%20a%20CAD%20systemy/CAD.pdf>
- [2] KALMAN, D.: *The Most Marvelous Theorem in Mathematics*. *Journal of Online Mathematics and its Applications*, ročník 8, 2008, [online], [cit. 2011-04-17]. URL <http://www.maa.org/joma/Volume8/Kalman/General.html>
- [3] KRŠEK, P.: *Základy počítačové grafiky – Studijní opora*. 2006-11-08, Fakulta informačních technologií, VUT v Brně.
- [4] KRŠEK, P.: *Vizualizace a CAD – Reprezentace dat v CAD systémech*. 2008, [online], Fakulta informačních technologií, VUT v Brně, [cit. 2011-04-15]. URL https://www.fit.vutbr.cz/study/courses/VIZ/private/lecture/viz_slide_cad_reprezentace_print.pdf
- [5] KRŠEK, P.: *Vizualizace a CAD – Sdílení dat v CAD systémech*. 2008, [online], Fakulta informačních technologií, VUT v Brně, [cit. 2011-04-15]. URL https://www.fit.vutbr.cz/study/courses/VIZ/private/lecture/viz_slide_sdileni_dat_print.pdf
- [6] KRŠEK, P.: *Vizualizace a CAD – Uživatelské rozhraní v CAD systémech*. 2008, [online], Fakulta informačních technologií, VUT v Brně, [cit. 2011-04-15]. URL https://www.fit.vutbr.cz/study/courses/VIZ/private/lecture/viz_slide_gui_print.pdf
- [7] KRŠEK, P.: *Vizualizace a CAD – Základy CAD systémů*. 2008, [online], Fakulta informačních technologií, VUT v Brně, [cit. 2011-04-15]. URL https://www.fit.vutbr.cz/study/courses/VIZ/private/lecture/viz_slide_zaklady_cad_print.pdf
- [8] MIŠKANIN, J.: *Význam CAD aplikací při vývoji nových produktů*. 2007, ISBN 978-80-8073-922-5, s. 397–400, [online], [cit. 2011-05-07]. URL <http://www.sjf.tuke.sk/novus/papers/397-400.pdf>
- [9] *Qt Reference Documentation*. [online], [cit. 2011-04-29]. URL <http://doc.qt.nokia.com/4.7/index.html>
- [10] *Qt Reference Documentation All Classes*. [online], [cit. 2011-04-29]. URL <http://doc.qt.nokia.com/4.7/classes.html>

- [11] Rao, P.: *CAD/CAM: principles and applications*. Mechanical engineering series, Tata McGraw-Hill, druhé vydání, 2004, ISBN 0-07-058373-0, [online], [cit. 2011-05-01].
URL http://books.google.com/books?id=QaY_sQC4-5YC
- [12] SHREINER, D.: *OpenGL programming guide: the official guide to learning OpenGL, versions 3.0 and 3.1*. Addison-Wesley, sedmé vydání, 2009, ISBN 978-0-321-55262-4.
- [13] *Tortoise SVN*. [online], [cit. 2011-04-29].
URL <http://tortoisesvn.tigris.org/>
- [14] VOJÁČEK, J.: *Matematika pro každého – Analytická geometrie*. [online], [cit. 2011-04-17].
URL <http://maths.cz/mapa-webu/analyticka-geometrie.html>
- [15] WEISSTEIN, E.: *Wolfram MathWorld*. [online], [cit. 2011-04-17].
URL <http://mathworld.wolfram.com/>

Dodatok A

Obsah CD

<code>\bin</code>	spustiteľné súbory
<code>\src</code>	zdrojové súbory aplikácie
<code>\doc</code>	zdrojové súbory technickej správy v \LaTeX u
<code>\doxygen</code>	dokumentácia zdrojových kódov
<code>2D_CAD_manual.pdf</code>	stručný popis rozhrania a použitia aplikácie vo formáte PDF
<code>bp.pdf</code>	technická správa vo formáte PDF
<code>readme.txt</code>	nápoveda k obsahu CD